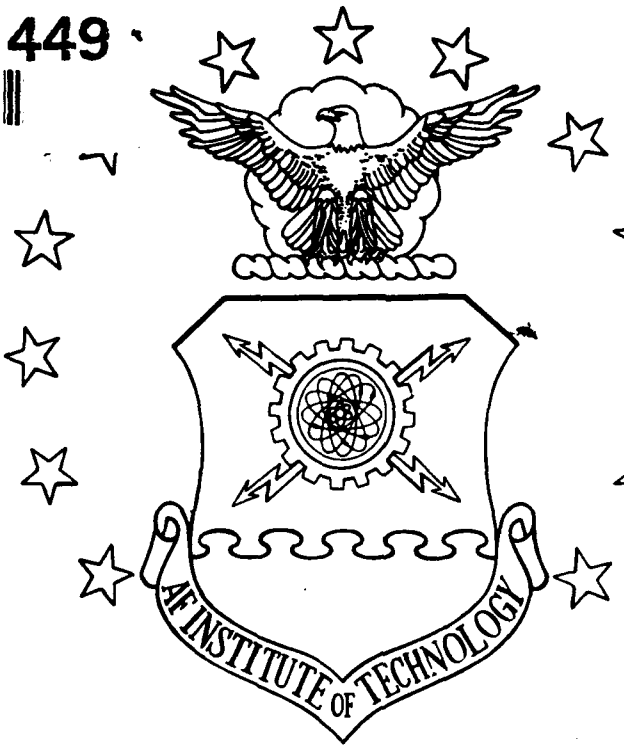

MULTISENSOR TARGET DETECTION AND CLASSIFICATION

THESIS

Dennis W. Ruck
Captain, USAF

AFIT/GE/ENG/87D-56

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

AFIT/GE/ENG/87D-56

MULTISENSOR TARGET DETECTION AND CLASSIFICATION

THESIS

Dennis W. Ruck
Captain, USAF

AFIT/GE/ENG/87D-56

AFIT/GE/ENG/87D-56

MULTISENSOR TARGET DETECTION AND CLASSIFICATION

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air University

In Partial Fulfillment of the

Requirements for the Degree of

Master of Science in Electrical Engineering

Dennis W. Ruck, B.S.E.E

Captain, USAF

December 1987

Acknowledgments

This research would not have been possible without the assistance of many people. I would like to acknowledge my indebtedness to them. Many thanks to my thesis advisor, Dr. Steven K. Rogers, for his unfailing encouragement, support, and advice. I would also like to thank my other committee members, Dr. Matthew Kabrisky and Dr. James P. Mills, for their assistance and encouragement throughout the research effort. Mr. Dave Powers from the Avionics Laboratory of the Air Force Wright Aeronautical Laboratories deserves my thanks as well, for he supplied me with the real world sensor data which makes the results of my work much more important. Finally, I would like to thank Dan Zambon, the systems engineer for the Information Systems Laboratory, whose help made the implementation and testing of my algorithms possible.

Dennis W. Ruck

Table of Contents

List of Figures

List of Tables

Abstract

In this thesis a new approach to the detection and classification of tactical targets using a multifunction laser radar sensor is developed. Targets of interest were tanks, jeeps, trucks, and other vehicles. Doppler images were segmented by developing a new technique which compensates for spurious doppler returns. Relative range images were segmented using an approach based on range gradients. The resultant shapes in the segmented images were then classified using Zernike moment invariants as shape descriptors. Two classification decision rules were implemented: a classical statistical nearest-neighbor approach and a new biologically-based neural network multilayer perceptron architecture.

The doppler segmentation algorithm was applied to a set of 180 real world sensor images. An accurate segmentation was obtained for 89 percent of the images. The new doppler segmentation proved to be a robust method, and the moment invariants were effective in discriminating the tactical targets. Tanks were classified correctly 86 percent of the time. The most important result of this research is the demonstration of the use of a new information processing architecture for military applications. The multilayer perceptron outperformed the nearest-neighbor classifier in every test.

A whole new generation of intelligent computers based on synthetic neural networks are now being investigated. This thesis establishes synthetic neural networks as a viable alternative to production rule-based expert systems for military computers.

MULTISENSOR

TARGET DETECTION AND CLASSIFICATION

1. Introduction

1.1. Historical Background

The problem of machine interpretation of visual images remains an unsolved problem. Since the advent of the digital computer, research in the area has produced many useful results although the general problem remains unsolved. Previous attempts at machine vision have utilized only a single sensor. The problem with using only a single sensor is that any sensor has certain weaknesses. These weaknesses reduce the robustness of the machine's vision. Currently, there is a great deal of interest in combining the information from several sensors to improve the robustness of machine vision. The goal is for the strengths of one sensor to offset the weaknesses of the other sensors, hence, improving the overall recognition capability of the machine. The Air Force is currently developing a multisensor device which combines passive visible, passive infrared, doppler laser radar, and relative range laser radar. The device is intended for use in detecting and identifying tactical targets of opportunity.

A former Air Force Institute of Technology (AFIT) student, C. Tong, worked on a technique for detecting man-made in the relative range imagery [Tong86]. This thesis will extend that work by 1) detecting targets in the doppler imagery and 2) classifying detections in the relative range and doppler images. The goal is to be able to detect and identify tactical targets in the relative range and doppler imagery.

1.2. Problem Statement and Scope

The general problem of interest is: Can a machine autonomously detect and classify man-made objects in an image? For this research project, the image was of a tactical scenario where the objects of interest were vehicles and all other objects such as buildings and trees were considered clutter. The problem consisted of two parts. The first part was to find the tactical targets in the image, a process known as segmentation; and the second part was to identify the targets as to the type of vehicle, a process known as classification. In this thesis, tactical targets will be detected and identified in the doppler and relative range imagery. The targets to be identified will consist of M60 tanks, Petroleum, Oil and Lubricant tankers (POLs), jeeps, 1.25 ton trucks, and 2.5 ton trucks.

1.3. General Approach

The detection and identification of targets will be performed in three steps. The first step is segmentation of the source images. The doppler images will be segmented using three new histogram-based techniques. The relative range images will be segmented using the technique developed in [Tong86]. The second step is the extraction of shape features from the segmented images. Zernike moment invariants will be used as rough shape descriptors. Finally, the extracted features will be classified using two different methods. Both a traditional statistical approach using a nearest-neighbor decision rule and a biologically-based neural network implemented as a multilayer perceptron will be used to perform classification. The multilayer perceptron is a network which can *learn* from past experience to improve its classification performance. The demonstration of the applicability of this new architecture for intelligent computers for Air Force relevant tasks is a major thrust of this thesis. The algorithms developed will be tested using actual sensor data which was collected in the field.

1.4. Thesis Organization

This chapter has provided a brief historical perspective on the problems of machine vision, a statement of the goal of this research effort, and an outline of the approach used to perform target detection and classification. The next chapter will review background material essential to understanding the algorithms which will be developed in chapter three. In chapter four, the results of applying the algorithms to actual sensor data will be discussed. Finally, conclusions and recommendations will be presented in chapter five.

2. Background Material

2.1. Introduction

In the last chapter, a brief historical background of the problems of machine vision was presented, and the exact problem to be solved was stated. This chapter will present background material essential to understanding the algorithms which will be developed in chapter three and tested in chapter four. The following topics will be covered in this chapter: 1) sensor and image database description, 2) segmentation techniques for doppler and relative range images, 3) shape description using moment invariants, 4) feature space reduction by Fisher linear discriminants, 5) classical decision rule techniques, and 6) neural network classification using multilayer perceptrons.

2.2. Sensor and Image Database Description

In this section, the source data used for design and test of the algorithms will be described. The source data was supplied by the Air Force Wright Aeronautical Laboratories Avionics Laboratory (AFWAL/AARI) and will be collectively referred to as the AFWAL database. Three main points will be covered: 1) general information about the image source data, 2) the variety of targets available in the source data, and 3) the subsets of the data used for classification tests.

2.2.1. General Information

The image source data was collected under contract by LTV Aerospace and Defense Company using their multifunctional laser radar (MFLR). For a given scene, the MFLR provides six images: doppler, passive infrared, passive visible, relative range, carrier intensity, and sideband intensity. The pixels are registered across all the images for a given scene. All images are 256 x 256 pixel images with eight bits per pixel.

The doppler image is encoded so that non-moving portions of the image are grey, objects moving towards the detector are lighter intensities (higher pixel values), and objects moving away from the detector are darker intensities (lower pixel values).

The relative range image is encoded so that objects farther from the detector have higher pixel values. The relative range images have a maximum unambiguous range of 15 meters [Target and Clutter Signatures from Multifunctional CO_2 Laser Radar, undated:7] so that objects more than 15 meters away have an apparent range equal to the actual distance modulo 15. For example, an object 16 meters away appears to be one meter away, hence, the term relative range image.

The carrier intensity image gives the strength of the received laser beam for each pixel in the image. Higher received carrier strengths result in higher pixel values. This image is significant when the laser beam is specularly reflected from the incident surface so that most of the beam is reflected away from the detector. In this case, the doppler images will contain spurious values where the carrier drops out. The other images were not used for this research, and therefore, will not be discussed.

2.2.2. Targets Available

The targets for which there were images in the source data are M60 tanks, jeeps, petroleum tankers (POLs), 1.25 ton trucks, 2.5 ton trucks, and radar shacks. However, the distribution of targets was heavily skewed towards tanks as shown in Table 2.1. This preponderance of tanks makes the task of testing classification algorithms very difficult. Ideally, one should have available an equal number of sample targets for each class with a uniform distribution of aspect angles within each class. Although such data was not available, it was possible to select a subset of the data for the purposes of classification. Two subsets of data were used for classification and will be described next.

Table 2.1 AFWAL Database Target Distribution

Class	#Samples
Tank	274
POL	99
Radar Shack	27
Jeep	10
Truck, 2.5 ton	8
Truck, 1.25 ton	8

2.2.3. Classification Subsets

Two subsets of the AFWAL database were formed for use in testing the classification algorithms. The first subset contained only three basic classes: 1) tank, front view, 2) tank, side view, and 3) POL, side view. Additional information regarding the small database is given in Tables C.1 and C.2 in Appendix C. The source images were selected on the basis that the objects of interest had sufficient shape definition to allow classification. Many of the images in the AFWAL database had very little shape definition due to the high detector to target range. In these images, even human observers could not reliably identify the classifications of the targets of interest; hence, the images were deemed unsuitable for attempting classification through shape analysis.

The second subset contained eleven classes. The eleven classes were derived from five vehicle classes by dividing each vehicle class into one or more sub-classes based on aspect angle. Specific information regarding this database is shown in Tables C.3 and C.4 in Appendix C. This database was deemed much more difficult for classification because many of the targets included showed significantly less shape definition than those in the small database. These less detailed images had to be included in order to obtain more

classes for testing the classification algorithms due to the small number of targets available other than tanks.

The different views of a given vehicle type were considered different classes because the shape descriptors employed were not aspect invariant; hence, aspects that differed significantly had to be considered different classes. For example, the side view of a POL differs significantly from the front view of a POL; hence, these two views constitute two different classes. In this research, the aspect angle was measured as the amount of counter clockwise rotation from the front view of an object about the vertical axis through the object. Hence, an aspect angle of zero degrees is the front view of an object, and an aspect angle of 90 degrees is the side view of an object with the frontal portion of the object on the viewer's right.

2.3. Segmentation Techniques

In this section, background to understand the segmentation techniques implemented in this thesis will be reviewed. The goal of segmenting an image is to identify all pixels within the image which represent the target. The output of a segmentation then is a binary mask where the target pixels are one and the background pixels are zero. Two segmentation techniques will be discussed. The first segmentation technique, optimum thresholding, is applicable to doppler images while the second method is applicable to relative range images.

2.3.1. Optimum Thresholding for Doppler Images

One method of segmenting an image is called optimum thresholding which is described by Gonzalez and Wintz in [Gonzalez77:325-331]. This method begins by assuming that there are two principal brightness regions in an image. If this is true, then a histogram of the brightness values in the image will be the sum of the histograms for the two regions in the image. For doppler images, one region is the background and the other is from the targets. Hence, the overall brightness histogram will have two modes,

one due to the background and the other due to the targets. Gonzalez and Wintz show that if the brightness distribution is known for each region, for example if the distributions are Gaussian, then it is possible to choose a threshold which minimizes the probability of incorrectly classifying a pixel. For the Gaussian case with the two distributions having equal variances, i.e., the noise is independent of which class is present, the optimum threshold is [Gonzalez77:327]

$$T = \frac{\mu_1+\mu_2}{2} + \frac{\sigma^2}{\mu_1-\mu_2} \ln\left[\frac{P_2}{P_1}\right] \tag{2.1}$$

where μ_1 and μ_2 are the means of the two distributions, σ^2 is the variance of both distributions, and P_1 and P_2 are the *a priori* probabilities of the two distributions. This method will be applied in the doppler image segmentation algorithm.

2.3.2. Relative Range Segmentation

A method for segmenting relative range images was developed by a former AFIT student, C. Tong [Tong86]. This algorithm assumes that the range gradients across targets will be relatively low while the range gradients for the background will be much higher. It also assumes that the target occupies a small portion of the overall image. With these assumptions, a histogram of the range gradients in the image will be a mixture of two distributions. The target distribution will be located at the lower range gradients and will have a small peak due to the small number of pixels on target compared to the number of background pixels. The background distribution will be located at the higher range gradients and have a large peak.

The segmentation algorithm is composed of five basic steps: 1) range gradient histogram enhancement, 2) thresholding based on the enhanced histogram, 3) conditional neighborhood filtering, 4) median filtering, and 5) density grow enhancement. Each step will be briefly discussed in order.

The range gradient histogram is first computed. Then the histogram is passed through a natural log transformation mapping function which emphasizes the lower range gradients.

Next a threshold is estimated from the enhanced range gradient histogram. The threshold is chosen so that it lies between the means of the target and background distributions. Once the threshold is set, the image is thresholded so that all pixels with range gradients less than the threshold value are labeled target pixels and all others are labeled background pixels.

The thresholded image is next conditionally filtered. The neighborhood around each pixel is checked. If the neighborhood does not support the current label for the pixel, then the label is changed. Thus, background pixels initially labeled target pixels will be converted to background pixels if neighboring pixels are labeled background. A relaxation method is used so that the neighborhood condition is dependent on the value of the range gradient for the current pixel.

After conditional filtering, a conventional median filter is used. The median filter eliminates speckle noise from the background and helps to fill in target pixels where the carrier signal dropped out.

Finally, a density grow algorithm is applied to the image. This algorithm helps to restore the edges of the object which are usually lost in prior processing stages without increasing the background noise. Tong showed this method to be effective in finding man-made objects in a relative range image. This technique will be applied in the segmentation of the relative range images. The next section discusses a set of features which can be used to provide shape discrimination of the segmented targets.

2.4. Shape Description Using Moment Invariants

2.4.1. Introduction

A variety of methods are available for shape description. The method used in this research effort was that of moment invariants. Moment invariants were chosen for two reasons. First, moments invariants are easily extended to provide greater shape discrimination as required. Second, moments can be computed optically. Optical computation is desirable in order to reduce the time required for the algorithm to detect and identify targets. The use of moments for shape description will be presented as follows: 1) definition of basic moments and invariants and 2) strengths and weaknesses of moment invariants.

2.4.2. Definition of Moments and Invariants

There are several types of moments that can be computed. The most basic moment is M_{pq} where

$$M_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q f(x,y)\, dxdy \tag{2.2}$$

If the function $f(x,y)$ is binary, then M_{pq} provides some information about the shape of the object, $f(x,y)$ [Pavlidis77:219]. However, these moments are not invariant under translation. In order to provide translation-invariance, central moments are used.

The central moments, μ_{pq}, are defined as follows

$$\mu_{pq} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} (x-\bar{x})^p (y-\bar{y})^q f(x,y)\, dxdy \tag{2.3}$$

where $(\bar{x},\bar{y})$ is the centroid of the image. These moments take on the same value regardless of where the object is located in the x–y plane [Pavlidis77:219]. However, these moments are not scale-invariant. In order to provide scale-invariance, normalized moments are defined.

The normalized moments, μ'_{pq}, can be defined in two different ways [Teague80:925]

$$\mu'_{pq} = \frac{\mu_{pq}}{(\mu_{00})^{(p+q+2)/2}} \tag{2.4}$$

or

$$\mu'_{pq} = \frac{\mu_{pq}}{(\mu_{20}+\mu_{02})^{(2+p+q)/4}} \tag{2.5}$$

where μ_{pq} is the centralized moment previously defined. The first method of normalization is equivalent to setting the total image area to unity; whereas, the second method is setting the radius of gyration to unity [Teague80:925]. Now the moments are translation and scale invariant. It is also desirable to have shape features which are rotationally invariant.

Rotationally invariant moments are more difficult to define due to the complex affect rotation has on the moments as traditionally defined. The effect on the moments of an object due to rotation through an angle of ϕ is [Teague80:925]

$$\mu''_{pq} = \sum_{r=0}^{j}\sum_{s=0}^{k}(-1)^{k-s}\binom{j}{r}\binom{k}{s}(\cos\phi)^{j-r+s}\times(\sin\phi)^{k+r-s}(\mu_{j+k-r-s,r+s}) \tag{2.6}$$

where μ''_{pq} is the moment of an object rotated through an angle ϕ and μ_{pq} is the centralized moment of the unrotated object. The scale normalized moment, μ'_{pq}, could also be used in equation (2.6) to determine the effect of rotation on these moments. Dudani used a set of rotationally invariant moments which were derived by Hu using this relation for the purpose of aircraft identification [Dudani77:41]. The moment invariants were derived using a combination of the central moments. Another set of rotationally invariant moments has been derived by Teague which are easier to generate for arbitrary order moments [Teague80]. These moments are based on the Zernike polynomials. Teague shows that the effect of rotation on the complex Zernike moments is multiplication of the

moment by a phase factor related to the amount of rotation and the degree of the moment [Teague77:926]. With this simple relation for rotation effects, Teague develops rotationally invariant combinations of moments up to the fifth order [Teague77:927-928]. The Zernike moment invariants can also be made invariant to changes in position and scale. It will be seen in the next section that the Zernike moment based invariants developed by Teague provide certain benefits over the moment invariants developed by Hu.

2.4.3. Strengths and Weaknesses of Moment Invariants

The use of moments for shape analysis suffers from several drawbacks; however, it will be seen that Zernike moments do not suffer from as many problems as the ordinary moments.

Yaser S. Abu-Mostafa and Demetri Psaltis developed a method for analyzing the capability of moments to perform shape analysis [Abu-Mostafa84]. They showed that moments generally suffer from three problems: 1) information loss, 2) information suppression, and 3) redundancy.

Information loss occurs when using moments because a finite number of moments can contain information regarding image variations up to a fixed spatial frequency which depends on the highest order moment used [Abu-Mostafa84:702]. Hence, information contained in the higher spatial frequencies is lost.

Information suppression occurs with moments because contribution to the moments is greatest near the edge of the unit circle when the image is normalized to be contained within the unit circle [Abu-Mostafa84:703]. Abu-Mostafa and Psaltis show that the weighting function for the moment calculation is as follows [Abu-Mostafa84:703]:

$$C_{(n+k)k} = 2\pi \int_0^1 r^{n+2k+1} c_{-n}(r)\, dr \qquad (2.7)$$

where C_{pq} is the complex moment and $c_{-n}(r)$ is the nth order circular harmonic expansion coefficient. From this relation, it is apparent that the portions of the image lying near

the edge of the unit circle will receive greater emphasis at the expense of the portions of the image lying near the origin. Hence, the information in the object near the origin is suppressed. The authors state that the complex moments, C_{pq}, are linear combinations of the ordinary moments, M_{rs}, satisfying [Abu-Mostafa84:698]

$$r+s=p+q \tag{2.8}$$

Thus, the ordinary moments suffer from information suppression. However, the authors state that the Zernike moments do not suffer from information suppression [Abu-Mostafa84:704].

Finally, Abu-Mostafa and Psaltis show that the ordinary moments suffer from information redundancy. They show that the ratio of new information to redundant information in the (n+2)th complex moment, $\rho(n)$, is [Abu-Mostafa84:704]

$$\rho(n)=\left[\frac{1}{(n+1)(n+3)}\right]^{1/2} \tag{2.9}$$

where $n=p+q$ is the order of the complex moment C_{pq}. Hence, as the order, n, of the moment increases the amount of new information provided approaches zero. However, the authors state that the Zernike moments do not suffer from information redundancy [Abu-Mostafa84:704]; hence, fewer Zernike moments than ordinary moments will be required to yield the same amount of information regarding the shape of an object.

After a set of features is calculated for an object, it is desirable to reduce the dimensionality of the feature space as much as possible while retaining the discrimination capability of the feature vectors. The next section will discuss one method for achieving this goal.

2.5. Fisher Linear Discriminants

The method of Fisher linear discriminants provides an effective tool for reducing the dimensionality of the decision space while maintaining the classification capability of

the feature vectors. This reduction in dimensionality is essential to make many classification algorithms practical. Typically, the dimension of the original feature space is much higher than the number of classes to be discriminated. The high dimensionality means that classification algorithms must perform a large number of calculations. If the number of calculations is very high, the method may be impractical for a given application such as guiding a bomb to its target where weight and space requirements limit the computational power available.

The method of Fisher linear discriminants is fully described in Duda and Hart's book *Pattern Classification and Scene Analysis* [Duda73:114-121]. The method will be briefly summarized here. For the complete mathematical description see [Duda73]. The method will first be described for the two class problem and then extended to the multiple class problem.

For the two class problem Fisher's method takes a feature vector and projects it onto a line in the feature space. The point on the line then determines to which class the vector belongs. If the original feature space is multidimensional, the classification problem will be much easier in the one dimensional space to which the feature vector maps. It will certainly involve fewer calculations. The difficulty lies in finding the correct line onto which to project the feature vectors while maintaining the separation of the classes.

Fisher's method finds the line which maximizes the ratio of the difference between the means of the two classes when mapped onto the line to the total scatter of the projected classes. The scatter for class i in the projected space is defined to be [Duda73:116]

$$\tilde{S}_i^2 = \sum_j (y_{ij} - \tilde{m}_i)^2 \qquad (2.10)$$

where y_{ij} is the jth sample from class i and $\tilde{m}_i$ is the mean of class i in the projected space. Note that the class scatter is the variance of the data for that class within a constant of proportionality in the projected space. The total scatter is defined to be the sum of the individual class scatters. Hence, the total scatter is proportional to the sum of the

variances of the projected classes. The goal is to minimize this sum while maximizing the distance between the means of the two classes when projected onto the line. Duda and Hart then proceed to show how to maximize the ratio of class separation to the total within-class scatter.

The extension to multiple classes is also treated by Duda and Hart in [Duda73:118-121]. In this case the mapping is from the original feature space to at most a $c-1$ dimension space where c is the number of classes. To see why at most $c-1$ dimensions are required consider the following. First, suppose that each transformed dimension gives information about whether the feature vector is in a class or not. Then $c-1$ discriminants are required because membership in the cth class is implied by nonmembership in the other $c-1$ classes; hence, only $c-1$ classes are required. However, less than $c-1$ dimensions may be used if a given discriminant gives information about more than one class. On the other hand, if two or more discriminants were required to give information about a single class, there would exist a combination of the dimensions which would yield information about the single class as in the two class problem already described; hence, at most $c-1$ dimensions are required.

Duda and Hart show how to generalize the two class problem to multiple classes. Again the process maximizes the separation of the means of the classes while minimizing the total within-class scatter [Duda73:120]. They show mathematically that the number of required discriminant functions is, indeed, at most $c-1$. Also, the transformation from the feature space to the reduced dimensionality decision space is not unique because the transformation could include rotations and scaling which would not affect the ratio that was maximized.

Once a feature vector has been derived for an object, it is necessary to compare that feature vector with a set of reference vectors of known objects. The next section will discuss the classical technique employed to perform the comparison.

2.6. Classical Decision Rule Techniques

There are several possible classical decision rules that can be implemented once the feature vector has been computed. Classical decision theory can be applied if sufficient information is known about the probability distributions that underlie the feature vectors. Decision theory can create decision rules which minimize the probability of error, minimize a risk function when errors are not all of equal importance (Bayes Risk), or assure a preset maximum false-positive (Neyman-Pearson).

For the problem at hand, however, the underlying distributions are not known, nor is it likely that they could be known. There are several reasons for the feature vector to fluctuate for a given object. Four reasons will be discussed below: aspect angle changes, target obscuration, range effects, and loss of received signal.

First, changes in the aspect angle to the object can cause severe changes in the feature vector. For example, the side view of a tanker truck will differ quite significantly from the front view resulting in a large change in the feature vector. Even when several views of an object are included in the set of reference vectors, the aspect angle changes, though not as dramatic, will still cause changes in the feature vector in an unknown manner. For example, if ten views of an object are included in the reference set, then aspect angle changes of ±18 degrees about each view will cause an unknown variation in the feature vector for that view which must be taken into account.

Another source of fluctuations in the feature vector for a given object results when parts of the object are obscured or lost. Trees, buildings, hills, other vehicles *etc.* can all cause parts of the object to be hidden from view. Also, imperfections in the segmentation algorithms used to detect the object may miss parts of the object. All these effects cause the feature vector to be changed in an unknowable way because the type of obscuration cannot be known in advance.

Changes in range can also cause changes in the feature vector. As the object moves away, less detail is available. The loss of detail will cause changes in the feature vector. However, careful selection of features should minimize this problem.

Low received signal at the detector can also cause portions of the object to be lost. Atmospheric conditions can result in sufficient attenuation so that the received signal is not detectable, or the surface conditions of the object may cause the laser beam to specularly reflect from the object away from the detector also resulting in loss of signal. The loss of signal will not be constant across the object, nor can the loss pattern be predicted; hence, the distortions caused cannot be predicted.

Since the probability distributions for the feature vector cannot be known, some approximations are required in order to apply classical decision theory for this case. Although the feature vector's probability distribution is not known, the distribution of the Fisher linear discriminants can be approximated by a Gaussian. Each component of the transformed feature vector is a sum of random variables; hence, the central limit theorem can be applied. The central limit theorem states that the normalized sum of independent random variables approaches a Gaussian distribution as the number of random variables approaches infinity [Ross76:257]. The normalized sum, Y, is given by [Ross76:257]

$$Y = \frac{\sum_{i=1}^{n}(X_i-\mu_i)}{\left\{\sum_{i=1}^{n}\sigma_i^2\right\}^{1/2}} \qquad (2.11)$$

where X_i is a random variable and μ_i and σ_i^2 are its mean and variance, respectively. Therefore, the Fisher linear discriminants will approach Gaussian random variables as the number of features increases.

For the case where the feature vectors have Gaussian distributions, classical decision theory yields the following results. First, it is assumed that the noise is independent of class; hence, the distribution for each class is the same except for the mean. A feature

vector then is given by

$$\vec{r} = \vec{s_k} + \vec{n} \tag{2.12}$$

where $\vec{s_k}$ is the mean for class k and $\vec{n}$ is the noise vector. For this case, the decision rule which minimizes the probability of error [Melsa78:111] is decide $\vec{r}$ is in class j if

$$\| \vec{r} - s_j \|^2 = \min_k \| \vec{r} - s_k \|^2 \tag{2.13}$$

In other words, decide $\vec{r}$ is in class j if it is closer to the mean for class j. This decision rule is called the nearest-neighbor decision rule. It is the simplest of decision rules.

The are many other classical and non-classical decision rules for classifying feature vectors. A non-classical decision rule employed in this research will now be discussed.

2.7. Neural Network Classification - Multilayer Perceptrons

The neural network approach attempts to mimic the behavior of the brain in order to perform classification. A neural network is a collection of simple computing elements (nodes) which are interconnected to form a network. The computing elements are similar to neurons in the brain in behavior (actually, the nodes are very simple abstractions of biological neurons). Figure 2. node gives the schematic for a node. The inputs to the node are either outputs from other nodes or the neural network inputs. The output of the node is given by [Lippmann87:13]

$$y = f\left[\sum_{i=0}^{N-1} w_i x_i - \theta \right] \tag{2.14}$$

where θ is the threshold for the node, N is the number of inputs to the node, x_i is the ith input to the node, w_i is the weight for the ith input, and $f(\cdot)$ is the output function which may be a hard limiting nonlinearity or a sigmoidal function. The type of output function used depends on the neural network being implemented. The way in which these nodes are interconnected and the interconnection weights determine the type of the neural net-

work. In this research, a multilayer perceptron was implemented [Lippmann87].

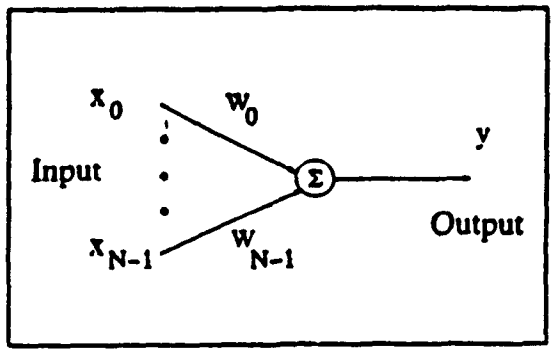

Figure 2.1 Schematic of a Single Node

The multilayer perceptron network is organized into three layers of nodes (see Figure 2.2). Each node in the first layer receives a weighted input from each of the components of the feature vector. The output of each node in the first layer is sent through a weighted connection to every node in the second layer. The interconnection between the second and third layer is the same as that between the first and second layer. Finally, the outputs of the third layer indicate the class of the input vector. The network is trained so that, for a feature vector from a given class, only one of the nodes in the third (output) layer is active and all other nodes are inactive. Thus, there is one node in the output layer for each class being detected. An existence proof by Kolmogorov states that a three-layer perceptron with $N(2N+1)$ nodes can compute any continuous function of N variables [Lippmann87:18]. Thus, the three-layer perceptron satisfying this condition can implement any required classification function. However, the proof gives no indication of how the weights should be set to achieve a given classification function [Lippmann87:18].

Rumelhart *et al.* recently popularized a method for training a multilayer perceptron [Rumelhart86]. The method is known as both the Generalized Delta Rule [Rumelhart86:322-328] and Back-Propagation [Lippmann87:17-18]. Back-Propagation was originally developed in 1974 by a Harvard PhD student, Paul J. Werbos, for his

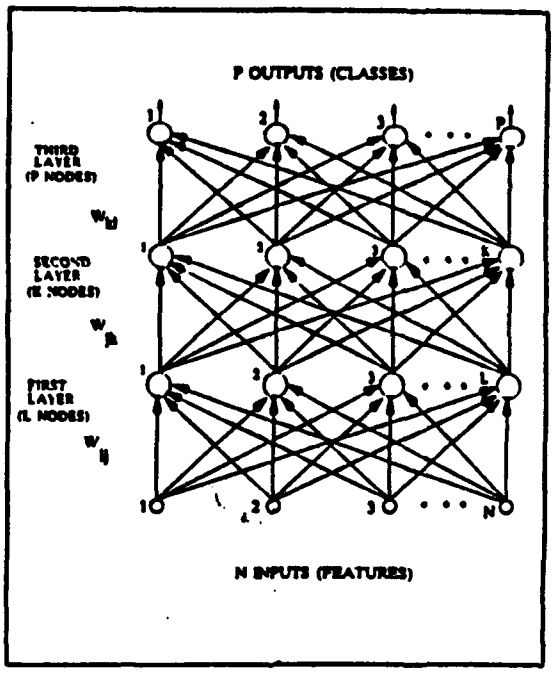

Figure 2.2 Multilayer Perceptron

thesis *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences* and was later rederived by David B. Parker in 1982 [Parker87]. The algorithm provides a way to set the weights in the multilayer perceptron network through a training process. First, a feature vector and the desired output for that input is presented to the network. The output function for a node is sigmoidal and is given by [Lippmann87:17]

$$f(\alpha) = \frac{1}{1+e^{-\alpha}} \tag{2.15}$$

The difference between the actual output of the network and the desired output is used as an error signal in adjusting the weights according to the equation [Lippmann87:17]

$$w_{ij}(t+1) = w_{ij}(t) + \eta\delta_j x'_i + \alpha(w_{ij}(t) - w_{ij}(t-1)) \tag{2.16}$$

where $w_{ij}(t)$ is the connection weight from node i in the previous layer or the ith input to node j in the current layer at time t, η is the training rate which is between zero and one, α is the momentum which is also between zero and one, x'_i is either the output of node i

in the previous layer or the ith input, and δ_j is the error term for node j. The error term for the jth output node is given by [Lippmann87:17]

$$\delta_j = y_j(1 - y_j)(d_j - y_j) \tag{2.17}$$

where y_j is the output of node j and d_j is the desired output for node j. For all internal nodes, the error term is given by [Lippmann87:17]

$$\delta_j = x'_j(1 - x'_j)\sum_k \delta_k w_{jk} \tag{2.18}$$

where x'_j is the output of node j and the summation is over all nodes in the next higher layer. Typically, the desired output has one node high and all other nodes low. The connection weights are updated in a recursive fashion starting at the output layer and working backwards toward the input layer as described above. The threshold for each node is similarly adjusted except that the threshold is assumed to be a weight on a connection to the node which has a constant input. Feature vectors from different classes are repetitively presented to the network until the weights and thresholds stabilize. After the network has been trained, the weights are fixed. To use the network for classification, a feature vector from an unknown class is input to the network. The output node that is highest indicates the class of the input feature vector.

This chapter has reviewed segmentation techniques for both the doppler and relative range imagery, a feature space dimensionality reduction technique, classical decision rule techniques, and multilayer perceptrons for classification. Each of these methods will be applied in the next chapter during the development of the segmentation and classification algorithms.

3. Segmentation and Classification Algorithm Development

3.1. Introduction

In the previous chapter, the background material essential to understanding the algorithms developed in this chapter was reviewed. The techniques examined will be applied in this chapter for the purpose of detecting and identifying targets in a tactical scenario. The overall algorithm consists of three main steps: 1) segmentation of the source image, 2) feature extraction from the segmented image using Zernike moment invariants as features, and 3) classification of the extracted features. The segmentation step, ideally, produces an image where all the target pixels have value one and the background pixels have value zero. The individual regions in the segmented region are located and the moment invariants are computed for each region. Finally, the set of moment invariants for a region is compared with the moment invariants of reference regions to determine the classification. The software used to implement the algorithms is described and listed in Appendix D. Each of the three major steps will now be discussed in greater detail.

3.2. Image Segmentation

The doppler and relative range images must be segmented to provide the shape of the targets in the scene for feature extraction. Three doppler image segmentation algorithms were employed, and a single method was used for performing relative range segmentation. The doppler image segmentation techniques will now be discussed.

3.2.1. Doppler Image Segmentation

3.2.1.1. Introduction

The first step in recognizing patterns in the imagery is to segment the areas of interest from the background and clutter. Segmentation of the doppler images was

achieved using three different techniques. Each technique will be described along with the impetus for using it.

3.2.1.2. Optimal Thresholding

The optimal thresholding technique depends on the probability distribution of an image being composed of two separate distributions, one for the target and one for the background. For the doppler images, this assumption should be true because the background will generally have a doppler value indicating no relative radial velocity while a radially moving target will have some nominal doppler value. Hence, the histogram for a doppler image should have two distinct humps. The optimal thresholding technique was applied to the doppler images using the assumption that the *a priori* probabilities of the target and background are equal. Although this is obviously not true, the wide variation possible for the probabilities makes it impossible to favor any pair of *a priori* probabilities. However, in special cases when these probabilities can be predicted, they should be used in the thresholding. With the above assumptions, the optimal threshold equation (2.1) which is based on Gaussian distributions becomes

$$T = \frac{\mu_t + \mu_b}{2} \qquad (3.1))$$

where μ_t and μ_b are the means of the target and background distributions, respectively. The means were estimated using the following procedure. First, the average of the histogram was computed. Since the histogram has two peaks in it, the average value should lie somewhere between the two peaks. Next the histogram was broken into two parts at the average value. The modes in each of the two parts were taken to be the means used in (3.1).

Once the threshold has been estimated, the image can be segmented. That is, each pixel with a value greater than the threshold takes one value while pixels less than the threshold take another value. After thresholding, the images are binary and can be

displayed in black and white. The polarity of the image will depend on whether the doppler value of the target was greater or less than the background value. Since it is desirable to have the resultant segmentation always have the same polarity, for example target pixels one and background pixels zero, the segmentation was further processed. It should be noted, however, that this simple technique will find only one group of targets when the source image contains targets moving both towards and away from the detector.

The final processing step guaranteed that the target pixels had the same value, namely one, independent of the relative velocity of the target in the doppler image. The polarity of the segmented image was reversed if the number of pixels having the value one was greater than half the number of pixels in the image. If the number of target pixels is always less than the number of background pixels, this process will result in the target pixels always having value one in the segmented image. For the data processed in this research, that assumption was always valid. A shorthand notation for this method to be used in figures and tables will be OT.

3.2.1.3. Optimal Thresholding Using Carrier Intensity

The above segmentation technique depends on the histogram of the doppler image being relatively free from noise. Unfortunately, this assumption is often not true. When the laser beam specularly reflects from the background away from the detector, there is a resultant loss of signal. In the doppler circuits, the loss of signal results in a spurious value being generated for that pixel. When the signal is lost for a large number of pixels, for example when specularly reflecting from a wet runway, the spurious doppler returns can invalidate the assumption of a bimodal distribution for the histogram. Hence, it is desirable to compensate for the loss of signal.

In order to compensate for the spurious returns generated by the loss of signal, the doppler image histogram was generated using the received carrier intensity image to determine whether or not a pixel should be included in the histogram. For each pixel in

the doppler image, the carrier intensity was checked. If the carrier intensity was below a threshold, the pixel was discarded. The value to use for carrier threshold was determined through examination of joint histograms of the doppler return value versus the carrier intensity for several images exhibiting spurious doppler returns.

For example, Figure 3.1 is a raw doppler image with significant spurious doppler returns which has been displayed using a greyscale encoding scheme. In the image, dark pixels represent low doppler return values, and lighter pixels represent higher higher doppler return values. Note the dark bands in the upper half of the image. These bands represent spurious doppler returns due to low received carrier intensity. Figure 3.2 is the joint histogram for that image. The horizontal axis of the joint histogram is received carrier intensity increasing to the right, and the vertical axis is the doppler value increasing towards the bottom of the image. The range of values for both axes is from zero to 255. The vertical line in the image represents a carrier intensity of 130. The histogram values are color coded as shown in the color bar at the bottom of the figure. Black is used only for a histogram value of zero. The other histogram values are mapped from blue through green to red. The histogram was normalized for display purposes so that the full color range would be utilized. Hence, the histogram values displayed are relative values. Except for the color black for a histogram value of zero, all other changes in color represent uniform steps in histogram values.

From the joint histogram, it is evident that spurious doppler return values can be eliminated with a carrier threshold of 130. Upon examination of several such histograms, it was determined that a carrier threshold of 130 would be sufficient in most cases. Once the doppler image histogram has been formed using the carrier intensity to discard spurious returns, the above segmentation technique can be applied. It was found, however, that this segmentation technique failed in many cases; hence, a better segmentation technique was sought. The Optimal Thresholding Using Carrier Intensity technique will be represented using a shorthand notation of OTUCI in subsequent figures and tables.

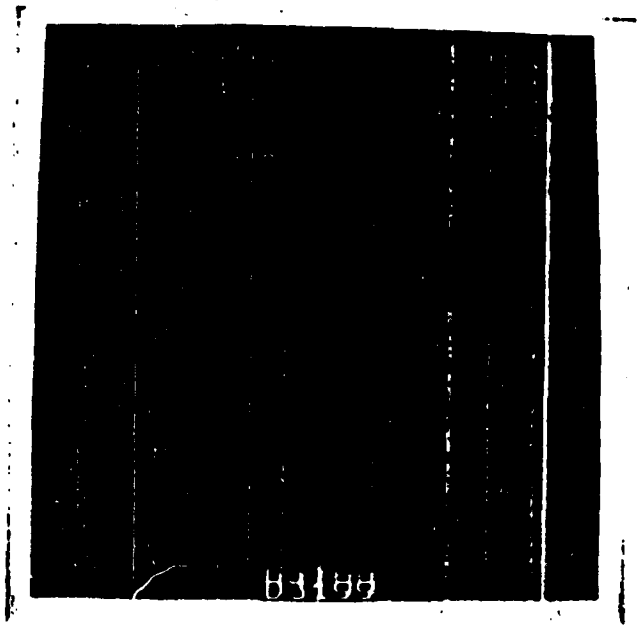

Figure 3.1 Raw Doppler Image Exhibiting Carrier Dropout. Note the dark bands in the upper half of the image which are due to carrier dropout.

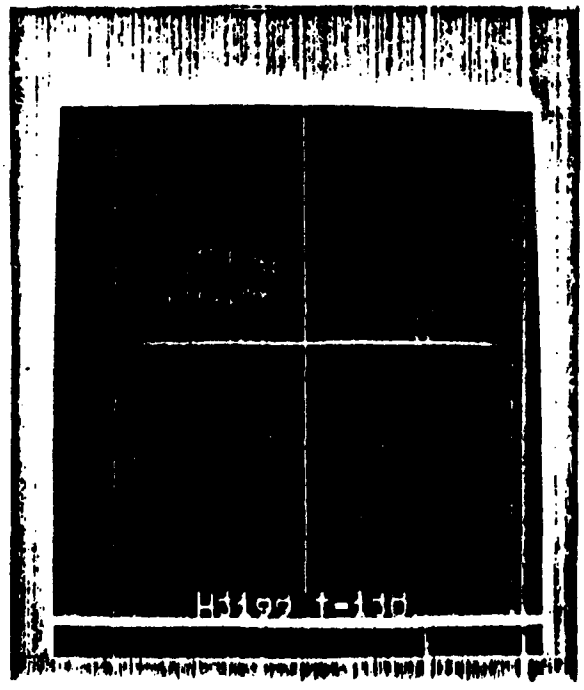

Figure 3.2 Joint Histogram between Doppler Return and Carrier Intensity. Note the clump of spurious doppler returns at low carrier intensities. The vertical line represents a carrier intensity of 130.

3.2.1.4. Background Lobe Elimination with Carrier Thresholding

The previous segmentation technique failed because for most images the background distribution dominated the histogram. Since the background dominated the histogram, the image average usually fell somewhere within the range of values representing the background. When the histogram was split at this point, the modes in each part of the histogram were due to the background. Hence, the target distribution had no effect on the threshold selected, and the target was many times not detected. Only when there was a sufficient number of pixels on target to weight the image average outside the main lobe due to the background was the target reliably found. Therefore, a method was developed which would eliminate this problem.

Since the main lobe in the histogram due to the background was causing the trouble, it was decided to eliminate the background from the histogram. The histogram was first constructed using the carrier thresholding described above to eliminate spurious doppler returns. Then the mode of the histogram was found. This mode represents the center of the main lobe of the background distribution. The histogram was then modified to eliminate all pixels due to the background. Starting with the mode of the histogram and working outward, the histogram bins were set to zero until the end of the main lobe of the background distribution was reached. The background lobe width was specified as a fractional value of the mode. Suppose the width was specified using 0.1 percent. Then all histogram bins starting at the mode working outward would be set to zero until the value of the histogram was 0.1 percent of the mode. Thus, the main lobe due to the background was eliminated. All segmentations performed with this method used a background lobe width specification of 0.1 percent.

Once the peak in the histogram due to the background was eliminated, the only peak left in the histogram was due to the target. Hence, the mode in the modified histogram was used as the mean of the target distribution, and the threshold was set to the average of the mode in the modified histogram, the target mean, and the mode in the original

histogram, the background mean. The image was then thresholded as described above. After thresholding, the segmented image was processed as described earlier to insure that the target pixels had value one and background pixels zero. The term BLE will be used as a shorthand notation in figures and tables for this method.

A limitation inherent in all three doppler segmentation techniques is that an image with multiple targets moving both towards and away from the detector will not be successfully segmented. The reason for this is that all three methods employ at some point a binary thresholding technique. An image with targets moving both towards and away from the detector should be segmented using at least three labels: moving away from the detector, background, and moving towards the detector. Since a binary thresholding technique uses only two labels, the pixels on the targets moving in the opposite direction of those targets successfully segmented will be incorrectly labeled background pixels because only two labels are available. A recommendation for future research is modification of the doppler segmentation methods to allow for images with targets moving in both directions.

These three techniques were employed in the segmentation of the doppler images. The relative range images were also segmented using an algorithm developed by a former AFIT student.

3.2.2. Relative Range Image Segmentation

All relative range images were segmented using the algorithm developed by C. Tong (GE-86D) in his thesis *Target Segmentation and Image Enhancement through Multisensor Data Fusion* [Tong86]. The computer program developed by Tong was used to perform the segmentations. A few changes were made to the software to reduce execution time; however, the algorithm was used essentially unchanged.

After the relative range and doppler images have been segmented, the next step is to compute the feature vector for each region in the segmented images. The next section will discuss this process.

3.3. Feature Extraction

The next step after segmenting the targets from the background in an image is to compute a set of features for the distinct regions within each image. This process consists of two parts. The first part creates a list of the distinct regions within the image, and the second part computes the feature vector for each region in the list. Each part will now be described.

3.3.1. Region Detection and Filtering

The purpose of this step is to create a list of all the distinct regions in an image. Each region will have attached to it sufficient information to allow the computation of the feature vector by the next process. This operation involves stepping up a level in the processing hierarchy. At the lowest level of image processing, only complete images are manipulated. Therefore, all operations must be applied to the image as a whole. At the next higher level of image processing, the image is partitioned into a set of separate entities. The image is the sum of the separate entities derived from the image; however, operations can now be defined which operate only on a single entity. This distinction is important because it has implications about the extent of operations which can be performed. For example, when operating at the image level, it is not possible to filter the image such that only objects above a certain size are passed. In order to perform such an operation, regions must be identified and processed separately. A similar operation in the image domain would be low-pass filtering of the image. That operation would eliminate small objects but would also distort the larger objects. On the other hand, processing each region in the image separately allows elimination of small regions while main-

taining, undistorted, the larger regions. Also, extracting regions from the image allows easier processing of the regions for purposes such as classification where the primary interest is in the specific region and not the entire image. By focusing attention on just the region of interest, considerable computations can be saved over processing of the entire image.

The algorithm developed extracted regions from the segmented image and filtered the regions to eliminate regions too small for classification purposes. The region extraction method will be described first followed by the region filtering method.

The goal of the region extraction algorithm was to produce a list identifying the distinct regions in an image. The region extraction consisted of two steps: 1) detection of a starting pixel for the next region and 2) creation of a list defining the boundary of the region. The starting pixel for the next region is found by scanning the image in a raster fashion until a pixel with value one is found that is not already included in the border of another region. Once the starting pixel is found, a list of the border pixels is created. The algorithm for creating the list of border pixels is as follows: starting from the initial border pixel follow the edge of the region counterclockwise by always trying to go to the right of the current pixel. As each border pixel is traversed, it is added to the list and is marked within the image as already belonging to a border. Marking the pixel within the image guarantees that the algorithm to find the start of the next region will not mistakenly flag a pixel that has already been traversed. After the list of border pixels for a region has been created, the region is added to the list of regions for the image. Once this list is created, it is filtered to eliminate regions which are too small for shape classification.

The region list for an image is next filtered to eliminate small regions. The algorithm counts the number of pixels on the border of each region and eliminates the region if the number of border pixels is below a preset threshold. The threshold was set to fifty pixels. This threshold value eliminates regions with approximately 200 pixels or less. A

square region of this size would have sides which are less than six percent of the image width; thus, the regions would be very small relative to the size of the entire image. It was deemed not necessary to perform shape classification on regions of this size.

The list of regions for an image is then passed to the next stage which computes the feature vector for each region in the list. The features computed will now be described.

3.3.2. Moment Invariants Computation

After a region has been detected in an image, the next step is to attempt to classify the region, for example as a tank, jeep, truck, or clutter. In order to classify the region, a set of features for the region must be calculated. The features are then compared with the features for known objects. The closest match between the unknown object and the known objects determines the classification of the region. The determination of which features allow the best discrimination is still an unsolved problem. For this research effort, it was determined to use the moments of the shape resulting from the segmentations. Since the goal is to discriminate tanks from jeeps and trucks which differ significantly in shape, it was determined to use rough shape descriptors. When only a few of the lower order moments are used, moments provide rough shape description. For example, the zero order moment

$$M_{00} = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x,y)\, dxdy \qquad (3.2)$$

where $f(x,y)$ is a binary function with value zero for the background and one for the target area represents the area of the target. The first order moments locate the centroid of the target, and the second order moments characterize the size and orientation of the target [Teague80:921]. The moments up to second order define an ellipse of a given size, eccentricity, and orientation [Teague80:921]. Hence, the moments are rough shape descriptors. Moments have been used before with success for shape discrimination [Dudani77]. Dudani used moments for identification of aircraft.

Zernike moment invariants were used for the calculation of shape descriptors for the detected regions. Zernike moment invariants provide a set of features which are position, scale, and rotation invariant (PSRI) [Teague80:927-928]. The moment invariants for each detected region were calculated twice. The first set of moments was calculated for the silhouette of the region, and the second set was calculated for the border of the region. When calculating the moments for the border, the function $f(x,y)$ took value one only for pixels on the border of the region and at all other points the function took value zero. This feature of calculating moments for both the border and silhouette was used by Dudani [Dudani77:42].

The Zernike moment invariants were determined for a region by first computing the centralized moments for the region according to equation (2.3) and normalizing for scale using equation (2.4). The resulting normalized central moments were used to compute the Zernike moment invariants using the formulas given by Teague [Teague80:929] (see Appendix E for the moment invariants computed). Normalized central moments up to the fourth order were used in computing the Zernike moment invariants.

A set of features have now been computed for the region which describe the shape of the region and are position, scale, and rotation (not aspect) invariant. These features form a feature vector which must be compared with the feature vectors for known objects in order to determine the classification of the unknown object. In the next section, two methods for performing this comparison will be discussed. Both methods were used in classifying the feature vectors.

3.4. Classification of Feature Vectors

Two approaches to classification of feature vectors were implemented. The first method utilized a classical approach applying a nearest-neighbor decision rule. The second method used a multilayer perceptron to perform classification of the feature vectors. The multilayer perceptron approach was implemented in order to demonstrate the

feasibility of using neural networks for classification. For each method, two sets of data were available. The first set consisted of feature vectors computed from a collection of training images. The purpose of having a set of training images is to characterize the expected variation of the feature vectors for each class of object to be recognized. The second data set consisted of features vectors derived from a collection of testing images. These vectors were used to test the classification accuracy of the decision rule derived from the training set. The classical approach will be outlined first followed by a discussion of the multilayer perceptron approach.

3.4.1. Statistical Classification

The statistical approach to classification of the feature vectors consisted of four steps: 1) normalization of feature vectors, 2) dimensionality reduction using Fisher Linear Discriminants, 3) calculation of class mean vectors, and 4) classification using a nearest-neighbor decision rule. Each step will now be discussed.

The feature vectors were first normalized in order to prevent differences in the scaling of the features to unduly influence the classification. For example, if one feature had most of its values a thousand times larger than any other features, then the distances between vectors would be dominated by that feature. The other features with smaller values would have little effect on the classification. To prevent this situation, the feature vectors were normalized as follows: first, the mean and standard deviation was computed for each feature across the entire training set. The ith component of each vector in the training set, x_i, was then transformed according to the equation below:

$$y_i = \frac{x_i - \mu_i}{\sigma_i} \tag{3.3}$$

where μ_i and σ_i are the mean and standard deviation, respectively, for feature i. The new training vectors $\vec{y}$ now have a mean vector of zero and standard deviation vector of unity; hence, the scaling for each feature is now identical. These computations were per-

formed using the Interactive Laboratory System (ILS) software package for signal processing. Appendix A discusses the mechanics of using ILS for this purpose. After normalizing the training vectors, the dimensionality was reduced using the Fisher Linear Discriminants.

The Fisher linear discriminants were computed for the training set in order to reduce the dimensionality of the feature vectors. The benefit of dimensionality reduction is that it lightens the computational load with minimal loss of discrimination capability [Duda73:117-118,121]. The Fisher linear discriminants were computed using ILS. Appendix A discusses how ILS was used to perform these computations. With the dimensionality of the feature vectors reduced, the class means were calculated.

The calculation of the class means is the third step in the classical approach to classification. The class means are used by the nearest-neighbor decision rule. The ILS software package was used to perform the computation of the class means as discussed in Appendix B. Once the means have been calculated, it is possible to perform classification testing using the test data set.

The final step in the classical approach is classification of test vectors using a nearest-neighbor decision rule. Each test vector was initially transformed according to equation (3.3). These transformed test vectors were then reduced in dimensionality using the previously calculated Fisher linear discriminants. Finally, the distance between a transformed test vector and each class mean was computed. The class mean with the smallest distance to the test vector determined the class of the test vector. Appendix B discusses the mechanics of these computations using ILS.

The above steps were used to classify feature vectors in the classical approach. Another method for classification using neural networks was applied to the feature vectors. The neural network approach will now be discussed.

3.4.2. Neural Network Approach - Multilayer Perceptrons

The second method used for classifying the features vectors was implemented using a multilayer perceptron. The multilayer perceptron was chosen for its ability to group into a single class distinct distributions in the feature space. For example, a POL will have distinctly different feature vectors for the front view and side view. For each view, the feature vectors will have some distribution about the mean for that class. The multilayer perceptron can implement a decision space where the separate distributions for the front and side view of a POL will result in the same classification [Lippmann87:14].

There are three main steps to using a multilayer perceptron for classification: 1) normalization of feature vectors, 2) training of the network, and 3) classification of test vectors. Each step will be discussed below.

The first step in implementing a multilayer perceptron for classification is normalization of the training vectors. The training set is normalized so that the training vectors span the unit hypercube with one corner at the origin. For each feature, the minimum and maximum value over the entire training set is computed. Next, the ith component of each training vector, x_i, is transformed according to the following equation:

$$y_i = \frac{x_i - \min_i}{\max_i - \min_i} \tag{3.4}$$

where $\min_i$ and $\max_i$ are the minimum and maximum, respectively, for feature i in the training set. This transformation guarantees that 1) for each feature all training vectors have values in the range [0,1], 2) there is at least one training vector where that feature has value zero, and 3) there is at least one training vector where that feature has value one. The purpose of this transformation is to ensure that all features have the same scaling and that the input vectors will be scaled similarly to the initial weights for the connections between nodes in the network. The normalization also places the inputs in the same range as the outputs of the network nodes. After the training vectors have been normalized, the network can be trained to classify those vectors.

The network is trained with the back-propagation method using equations (2.16), (2.17), and (2.18) and the feature vectors contained in the training set. The weights and thresholds in the network are initialized randomly using a uniform distribution between -0.5 and 0.5. The desired output vector has value 0.9 for the element corresponding to the class of the input vector and all other elements are 0.1. The values 0.1 and 0.9 are used instead of 0 and 1 because the sigmoidal function on the output nodes [see equation (2.15)] can only reach zero and one for infinite negative and positive inputs, respectively. Forcing the output to zero and one would cause the weights to the output nodes to become very large [Rumelhart86:329]. The network is trained until the distance between the desired output and the actual output drops below a preset threshold. This training method will drive the outputs to 0.1 and 0.9; however, a modification should be investigated where if the output is greater than 0.9 for the node which is supposed to be active, then do not drive the output toward 0.9; and if the output is less than 0.1 for a node which is supposed to be inactive, do not drive the output toward 0.1. This modification would allow but not force the outputs to zero and one. The result may be better performance than that achieved using the present method of training. After the network has been trained, it can be used for classification.

The trained multilayer perceptron was then used to perform classification of the feature vectors in the test set. The feature vectors were transformed according to equation (3.4) and input to the network. The output of the network was examined to determine classification. There are several ways to determine classification from the output vector. For example, the maximum output could simply determine the class. A different technique was used in this project. Two thresholds were set: a minimum within-class threshold and a maximum out-of-class threshold. For a network output to be unambiguously classified, it was required that a single element in the output must be above the within-class threshold and all other elements must be below the out-of-class threshold. If this condition was not met, the output was declared ambiguous.

This chapter has outlined the algorithms developed during the research to perform tactical target detection and classification using doppler and relative range images. The results of applying this set of algorithms to the AFWAL database will be discussed in the next chapter.

4. Segmentation and Classification Results

4.1. Introduction

In the last chapter, algorithms were developed for the segmentation and classification of doppler and relative range images. These algorithms were applied to actual sensor data from the AFWAL database to test their effectiveness. The results will be discussed in two parts. The first part will present the results of applying the segmentation algorithms including a discussion of aspects of the results pertaining to the task of classifying objects in the image. The second part will discuss the results of the two classification methods applied to the segmented images. Two of the new doppler segmentation algorithms successfully segmented over 87 percent of the images tested. Both of the classification methods demonstrated were successful; however, the lack of sufficient test data limits the conclusiveness of the results. The segmentation results will be presented first followed by the classification results.

4.2. Segmentation Results

In this section, the results of applying the three doppler segmentation techniques and the relative range segmentation algorithm to the AFWAL database will be discussed. The effectiveness of the segmentation techniques were evaluated by a human observer.

The observer examined the source image for a given scene and the segmented doppler or relative range image and judged the segmentation effectiveness on two aspects: 1) the percentage of pixels on target detected and 2) percentage of background pixels falsely labeled target pixels (noise). For the target rating, the following codes were used: *excellent* = all target pixels detected; *good* = approximately two-thirds of the target pixels detected; *fair* = approximately one-third of the target pixels detected; and *poor* = none or very few target pixels detected. For the noise rating, the following codes were used: *excellent* = no noise pixels; *good* = approximately 100 noise pixels; *fair* =

approximately 1,000 noise pixels; and *poor* = several thousand noise pixels resulting in large contiguous noise regions. The doppler segmentation results will now be presented using this rating system.

4.2.1. Doppler Segmentation Results

The doppler images were segmented using three separate algorithms. The results of applying each algorithm to the doppler images will now be discussed. The doppler images analyzed are listed in Tables C.5, C.6, C.7, and C.8 of Appendix C. A total of 180 images were analyzed using each of the three methods. The images were selected to have a wide variety of target sizes, speeds and background noise.

4.2.1.1. Optimal Thresholding Results

The first segmentation technique applied to the doppler images was the optimal thresholding technique (OT) described in Chapter 3. The results are summarized in Tables 4.1 and 4.2. As shown in the table, this method of segmentation performed very well in the majority of cases. For 87.2 percent of the images analyzed, the identification of target pixels was *good* or better. Noise immunity was *good* or better for 77.7 percent of the images analyzed. Figure 4.1 is an example where the segmentation was judged *excellent-good*. Figure 4.2 shows the source image using a special display method to highlight the target. Note that nearly all target pixels have been properly identified and that noise pixels are non-existent. Figure 4.3 is a segmentation which was judged *poor*. Figure 4.4 is the source image displayed using a greyscale coding scheme. In this case, all the target pixels were missed, and there is a large number of noise pixels in the image.

The twenty-two images in which the targets were not located were characterized by moderate to severe carrier dropout. That is, specular reflection of the laser beam away from the detector resulted in a loss of signal and concomitant spurious doppler values. Figure 4.4 shows a case where the beam is being specularly reflected from runways in the scene yielding spurious doppler return values. Figure 4.5 is a joint histogram of the

Table 4.1 Doppler Segmentation - Target Location

Quality	OT (%)	OTUCI (%)	BLE (%)
Excellent	84.4	0.0	71.1
Excellent-Good	0.6	0.0	7.2
Good	2.2	43.3	10.6
Good-Fair	0.0	1.7	3.3
Fair	0.0	1.1	3.9
Fair-Poor	0.6	0.6	0.6
Poor	12.2	53.4	3.4

Table 4.2 Doppler Segmentation - Noise Immunity

Quality	OT (%)	OTUCI (%)	BLE (%)
Excellent	64.4	66.7	92.2
Excellent-Good	0.0	1.1	1.7
Good	13.3	25.6	4.4
Good-Fair	2.8	2.2	0.0
Fair	3.9	1.7	1.7
Fair-Poor	2.2	2.2	0.0
Poor	13.4	0.6	0.0

doppler return value versus the carrier intensity. As can be seen, a large number of pixels in the image have spurious doppler values. The computation of the image average for this image results in a value between the zero motion doppler value and the spurious

Figure 4.1 Doppler Segmentation of Scene 3033 Using OT. Compare this segmentation with the raw doppler in Figure 4.2. This segmentation was judged *excellent-good*.

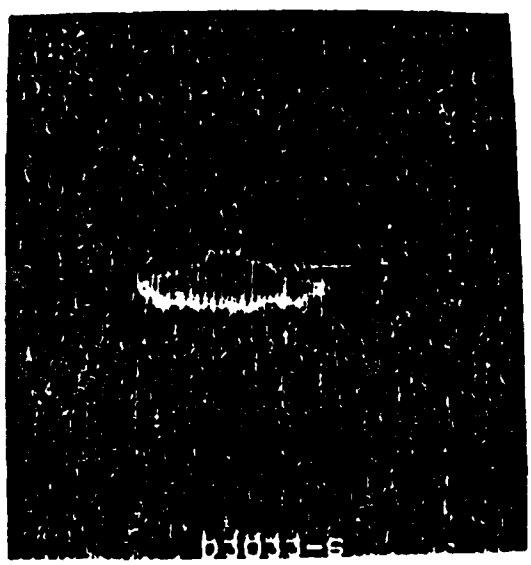

Figure 4.2 Raw Doppler of Scene 3033 Using a Special Display. Compare with the segmented image in Figure 4.1.

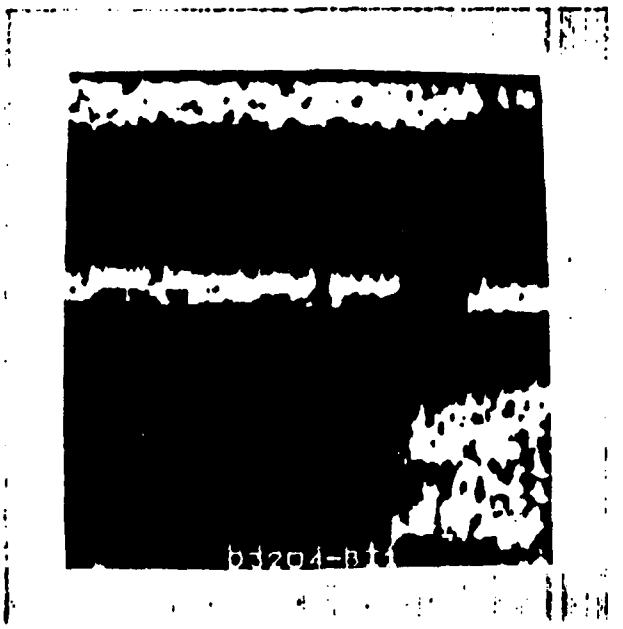

Figure 4.3 Doppler Segmentation of Scene 3204 Using Optimal Thresholding. Compare the segmentation with the raw doppler in Figure 4.4. This segmentation was judged *poor*.

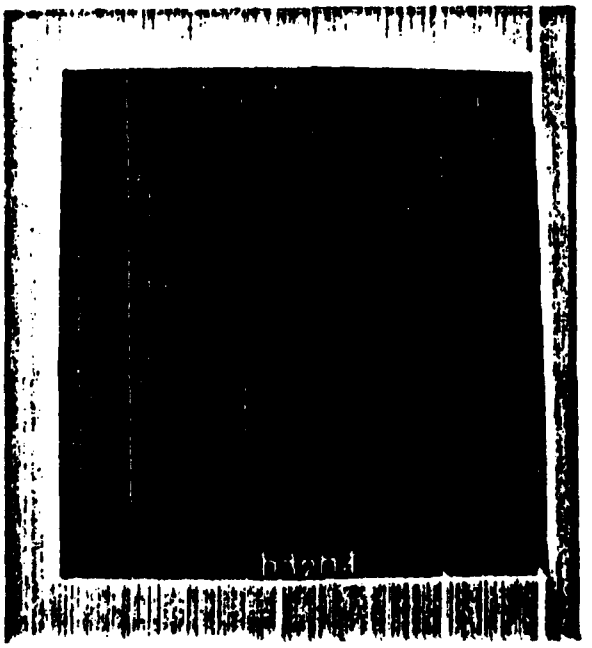

Figure 4.4 Raw Doppler Image of Scene 3204. Compare with the segmented image in Figure 4.3.

doppler value. Hence, the resultant segmentation groups the target pixels with the background and the spurious doppler returns are identified as target pixels. Figure 4.3 shows the resultant segmentation. Obviously, there is a need to compensate for the presence of carrier dropout during the segmentation process. The second method of segmentation was developed for this purpose.

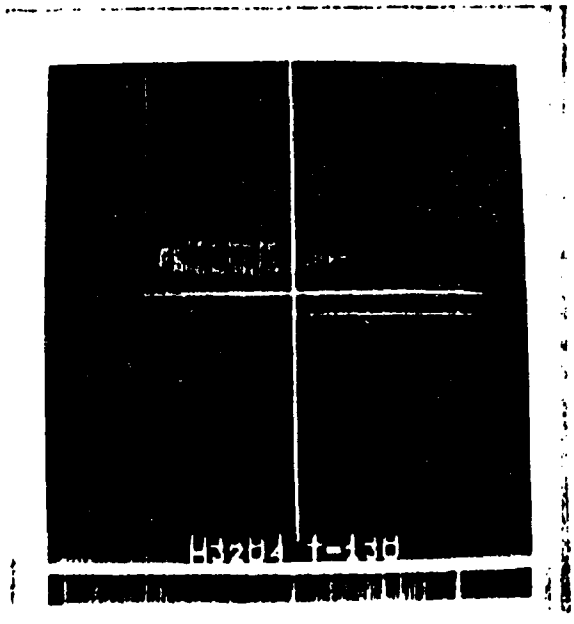

Figure 4.5 Joint Histogram for Scene 3204. Note the large number of spurious doppler values.

4.2.1.2. Optimal Thresholding using Carrier Intensity Results

The second method of segmentation, optimal thresholding with carrier dropout compensation (OTUCI), was intended to provide accurate segmentations in the presence of carrier dropout. The results of this segmentation method are summarized in Tables 4.1 and 4.2. This method of segmentation received at best a target rating of *good* for any segmentation. The reason for this is that all the segmented targets had small holes in the interior due to carrier dropout on the target. Figure 4.6 shows the raw doppler for a scene with severe carrier dropout using a special display method to highlight the targets. The mode of the image was displayed black, and all other doppler values were mapped

normally from black to white. Figure 4.7 shows the raw doppler using the normal display method. Note that the targets are barely visible because their doppler returns differ very little from the background; hence, the contrast between the targets and the background is very low. Figure 4.8 shows the result of segmenting the doppler image. Note that there is little noise in the segmentation due to carrier dropout: noise is effectively eliminated by this method. Also, the method missed one of the targets because it was moving in the opposite direction of the primary target. Figure 4.9 shows the resultant segmentation using the first method (OT). Note that the targets were lost completely.

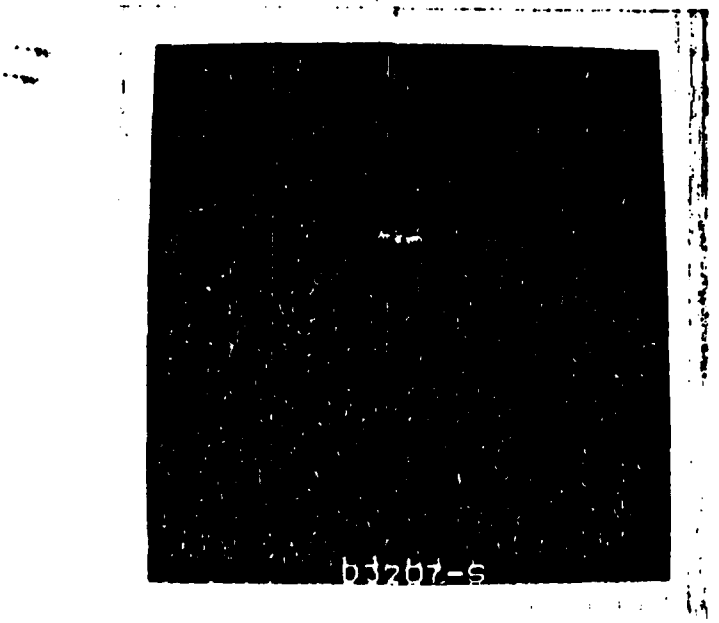

Figure 4.6 Raw Doppler of Scene 3207 Using a Special Display Method. Compare with Figure 4.7. Note the enhancement of targets using this display method.

From Table 4.1 it is apparent that the method failed to detect the presence of the target for more than half of the images analyzed. Although the noise performance was quite good, the failure to detect more than half of the targets required modification to the method to improve the detection of targets while maintaining noise immunity.

The problem with detection of targets stemmed from the fixed threshold used for the carrier and its application to all doppler returns. For some doppler images the

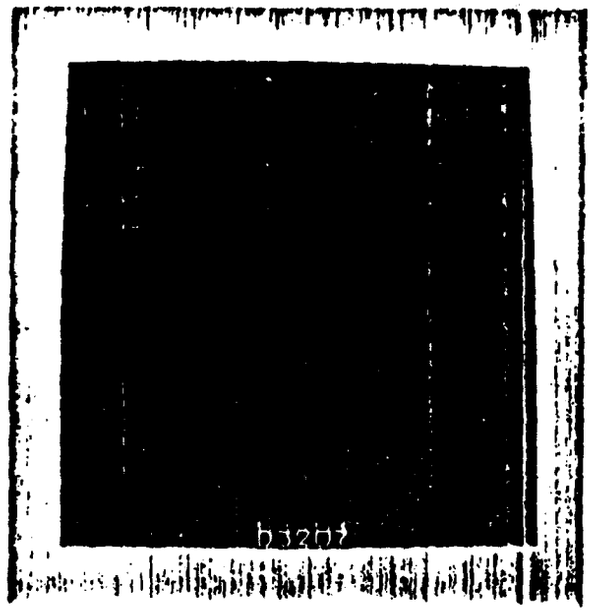

Figure 4.7 Raw Doppler of Scene 3207. Compare with Figure 4.6. Note the difficulty of identifying targets in the raw doppler image.

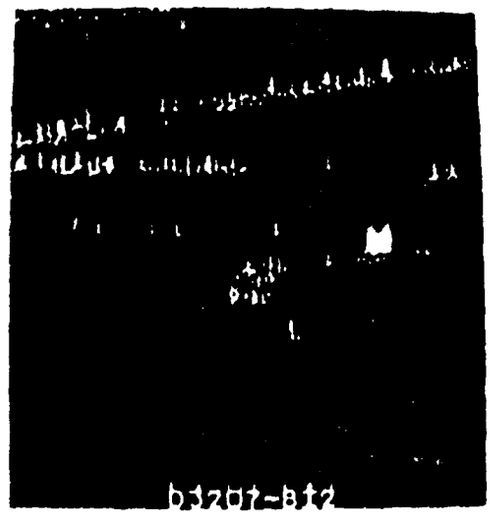

Figure 4.8 Doppler Segmentation of Scene 3207 Using OTUCI. Note that a target was found in the presence of carrier dropout and the amount of noise in the segmentation is low.

Figure 4.9 Doppler Segmentation of Scene 3207 Using Optimal Thresholding. Note that no targets were detected using this method.

spurious doppler returns had carrier intensities above the 130 threshold as shown in Figures 4.4 and 4.5. Since the spurious doppler values are very far from the zero motion value, they have a large effect on the computation of the image average. Also, a large number of the zero motion pixels are eliminated because the associated carrier intensity is below the threshold. Elimination of these background pixels makes it easier for the image average to be swayed by the spurious doppler returns. The result is that even when there are only a few spurious doppler returns with carrier intensity above the threshold, they overpower the genuine target returns resulting in a poor segmentation as shown in Fig. 4.10. The third doppler segmentation technique was developed to overcome these problems.

4.2.1.3. Background Lobe Elimination with Carrier Thresholding Results

The Background Lobe Elimination with Carrier Thresholding method (BLE) was developed to provide effective detection of targets in doppler images in the presence of spurious doppler returns due to carrier dropout. The results of applying this method to

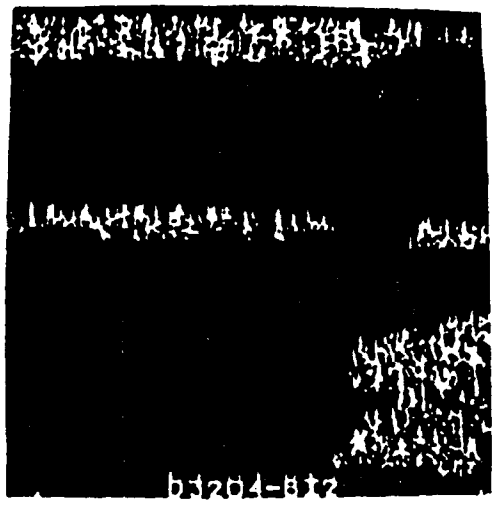

Figure 4.10 Doppler Segmentation of Scene 3204 Using OTUCI. Note the failure to detect the targets due to the presence of carrier dropout.

the doppler images are summarized in Tables 4.1 and 4.2. This method of segmentation is better than either of the two previously discussed methods (see Table 4.3 for a comparison). The target location performance is slightly better than the first method and much better than the second method while the noise performance is much better than the first method and somewhat better than the second method. This method failed to find some portion of the targets in an image only 3.4 percent of the time compared to 12.2 percent for the next best method (OT). Figure 4.11 is the segmentation resulting from applying this method to Figure 4.4. The method successfully detected the targets in the presence of severe carrier dropout. However, this method is not without its weaknesses.

The primary weakness of this method of segmentation results from the merging of the background lobe and the target lobe. When there is a large spread of doppler return values from the target near the zero motion value, portions of the target are lost. Figure 4.2 is the raw doppler displayed with the background set to black, and Figure 4.12 is the resultant segmentation using this method. Note from the joint histogram (Fig. 4.13) that

Table 4.3 Doppler Segmentation Comparison

	OT (%)	OTUCI (%)	BLE (%)
Target Location Good or Better	87.2	43.3	88.9
Noise Immunity Good or Better	77.7	93.4	98.3

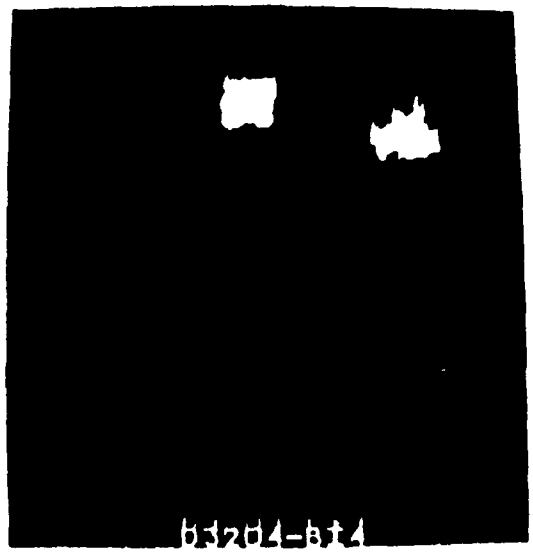

Figure 4.11 Doppler Segmentation of Scene 3204 Using Background Lobe Elimination. Note the targets were detected in the presence of severe carrier dropout.

the target return values are ntiguous with the background (zero motion) values. When the algorithm removes the background lobe, only the fastest moving portions of the target are retained.

A weakness of all three segmentation methods is the inability to detect in a single image targets which are moving both towards and away from the detector. Since all three methods perform at some point a binary thresholding of the image, either the

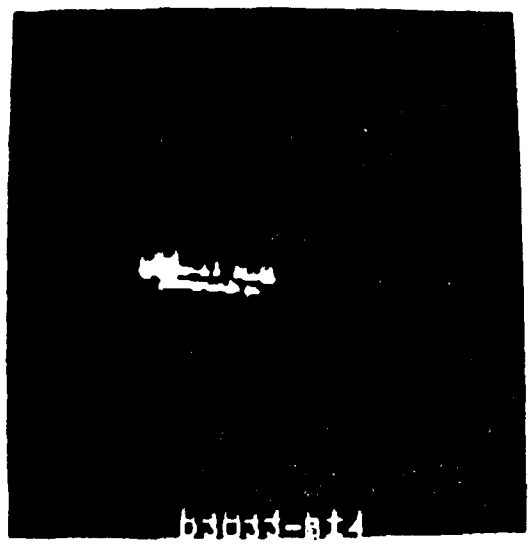

Figure 4.12 Doppler Segmentation of Scene 3033 Using Background Lobe Elimination. Note the failure to detect all portions of the target.

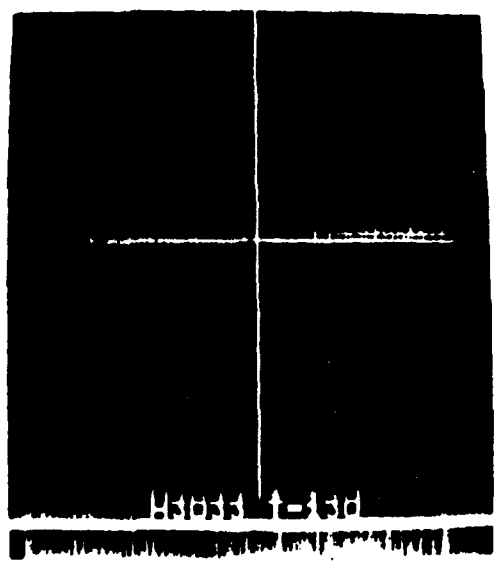

Figure 4.13 Joint Histogram of Scene 3033. Note that the target return values are contiguous with the background.

targets moving toward or away will be merged with the background. The targets in such an image which are found will be those with the greater number of pixels; hence, the

final segmentation will show the largest targets in the field of view. The targets moving in the opposite direction which are lost will have fewer pixels; hence, these targets are smaller. Future research should include modification of these methods to eliminate this weakness.

In summary, it was found that the Optimal Thresholding method worked well except in the presence of carrier dropout. The method located most targets and produced little or no noise. The Optimal Thresholding using Carrier Intensity method reduced the noise due to specular reflection significantly but failed to detect targets over 50 percent of the time. The Background Lobe Elimination with Carrier Thresholding method was superior to either of the two methods. This method was immune to noise due to specular reflection and found the targets for all but the most difficult cases. The results of the relative range segmentation algorithm will be discussed next.

4.2.2. Relative Range Segmentation Results

The algorithm developed by C. Tong (GE-86D) for the segmentation of relative range images was tested on 74 images. The algorithm was successful at finding flat targets within the images. Table 4.4 summarizes the segmentation results. Figures 4.14 and 4.15 show the original relative range image and the final segmented image as an example of the performance of this technique.

The algorithm has three main weaknesses: 1) splitting of targets, 2) noise from low range gradient backgrounds, and 3) segmentation of buildings. Figures 4.16 and 4.17 exemplify the problems listed. Note that the turret of the tank has been separated from the body of the tank. Target splitting is a serious problem for classification algorithms based on the silhouette of objects because, for example, the shape of a split tank is radically different from an unsplit tank. Also note in Figure 4.17 that the foreground has been included in the segmentation because it has a sufficiently low range gradient for the algorithm to identify it as a target. Finally, observe that the buildings in the upper portion of

Table 4.4 Relative Range Segmentation - Target Location and Noise Immunity

Quality	Target Location (%)	Noise Immunity (%)
Excellent	0.0	6.8
Excellent-Good	1.4	6.8
Good	29.7	36.5
Good-Fair	13.5	12.2
Fair	33.8	23.0
Fair-Poor	4.0	9.4
Poor	17.6	5.4

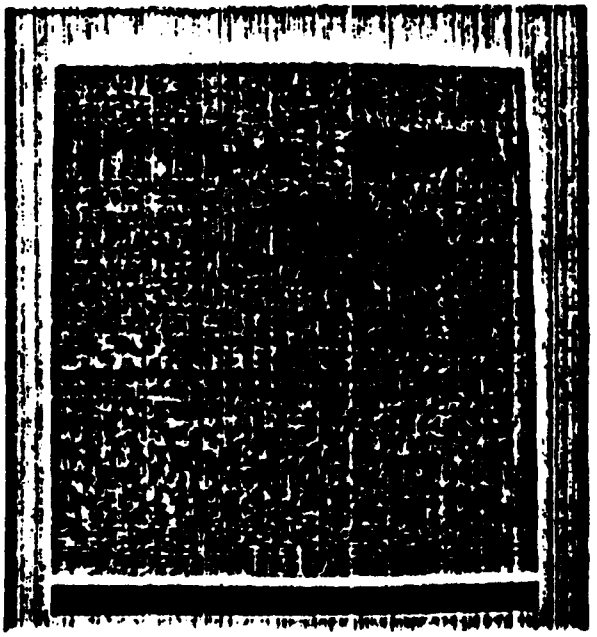

Figure 4.14 Relative Range Image for Scene 3195

the image have been included in the segmentation. Although in general that may not be a problem, it is a problem here because only vehicles are being sought (it should be noted, however, that Tong developed this algorithm for the detection of man-made objects which included buildings). Except for the low range gradient background images, the

Figure 4.15 Relative Range Segmentation of Scene 3195

segmentations were generally free of spurious regions.

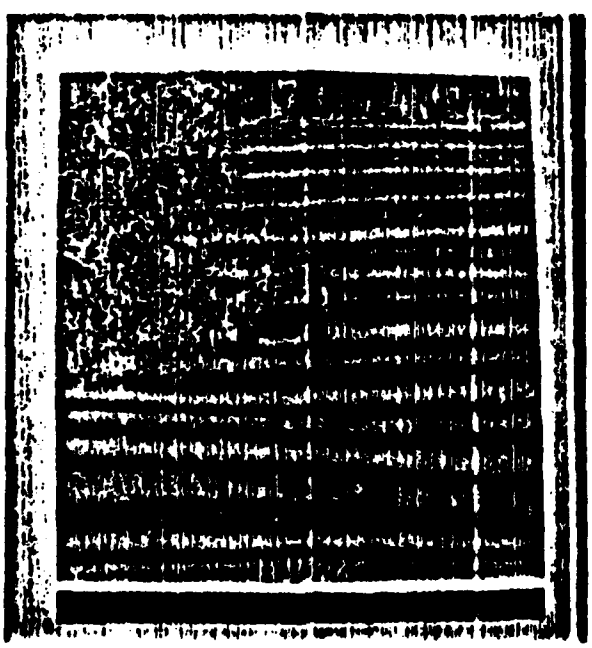

Figure 4.16 Relative Range Image of Scene 3026

The segmented images were then processed to identify the distinct regions and compute the Zernike moment invariants. These feature vectors were then used to classify the

Figure 4.17 Relative Range Segmentation of Scene 3026. Note that 1) the tank has been split into two parts, 2) the noise in the foreground due to the low range gradient, and 3) the buildings have been included in the segmentation.

segmented regions. The results of classification using this approach will be discussed in the next section.

4.3. Classification

After the source images were segmented, the moment invariants were computed for all distinct regions within the images. These feature vectors were then classified using two different methods: a nearest-neighbor decision rule and a multilayer perceptron. The results of using the nearest-neighbor decision rule will be presented first followed by the results for the multilayer perceptron.

4.3.1. Nearest-Neighbor Classification Results

A classical approach to classifying the feature vectors was implemented using a nearest-neighbor algorithm after reducing the dimensionality of the feature vectors using Fisher linear discriminants. As stated earlier, two subsets of the AFWAL database were

used for the classification tests. The results for each subset will be discussed separately. Also, a test was performed to guarantee that the feature vectors being calculated were position, scale, and rotation invariant (PSRI). This test will be discussed first.

4.3.1.1. PSRI Test of Feature Vectors

A test of the feature vectors calculated was performed to insure that the features were invariant under changes in position, scale, and rotation of the object. For this purpose, three objects were selected: 1) front view of a tank, 2) side view of a tank, and 3) side view of a POL. From each object, several new objects were generated which differed from the original by either a change in position, scale, or rotation. For position changes, five different positions were used: 1) centered, 2) shifted up and to the left, 3) shifted up and to the right, 4) shifted down and to the left, and 5) shifted down and to the right. The amount of shift was one fourth of the image height. For scale changes, nine different scales were used from 1.0 to 0.2 by steps of 0.1. For rotation changes, eight different rotations were used from zero degrees to 315 degrees by steps of 45 degrees. The direction of rotation was counter clockwise for positive rotations.

The feature vectors were found to be relatively invariant under changes in position, scale, and rotation. The results of the test will be displayed in scatter plots. A scatter plot is generated by performing the feature space dimensionality reduction (Fisher linear discriminants) and plotting the first two coordinates of each transformed feature vector. Since there are only three classes, the reduced feature space is guaranteed to be at most two dimensional. Figure 4.18 is the scatter plot for the reduced feature vectors when the scale of the objects was changed; Figure 4.19 is the scatter plot when the rotation of the objects was changed; and Figure 4.20 is the scatter plot for the reduced feature vectors of all the objects generated including position, scale, and rotation changes as well as the original objects. A scatter plot could not be generated using the objects which differed only in position because the software used to compute the Fisher linear discriminants could

not calculate the transformation when the within-class scatter was zero. The feature vectors were manually verified to be identical when only the position of the object was changed.

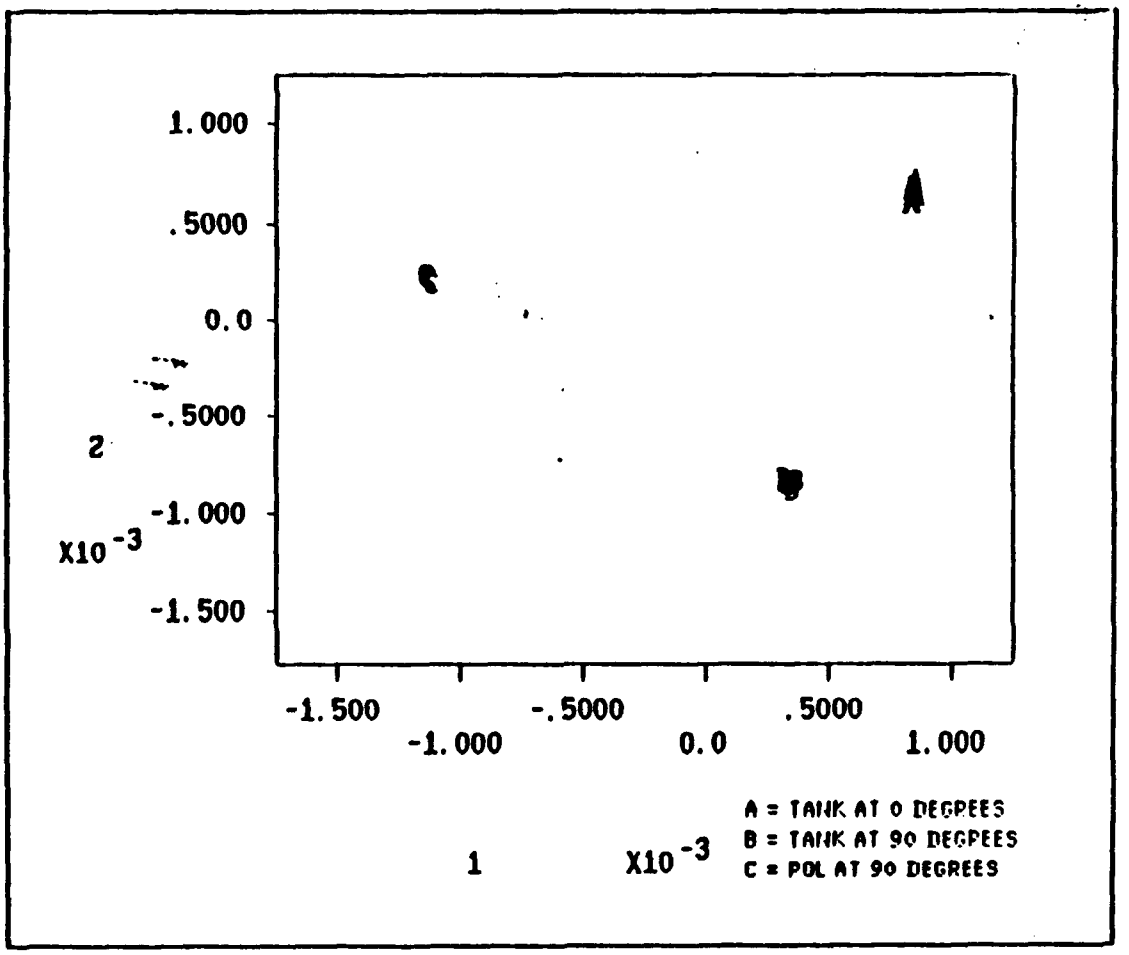

Figure 4.18 Scatter Plot of Feature Vectors for Scale Changes

From the scatter plots, it is evident that the feature vectors are not perfectly invariant under the operations of scaling and rotation. The reason for this is that an object is represented by a discrete set of pixels in a plane. When the object is rotated or scaled, the boundary pixels are moved to new locations in the image plane where there may or may not be a pixel location. If there is no pixel location for the new point, the nearest pixel location must be selected. This approximation of the rotated or scaled boundary results in minor fluctuations in the feature vectors for the scaled and rotated objects. This

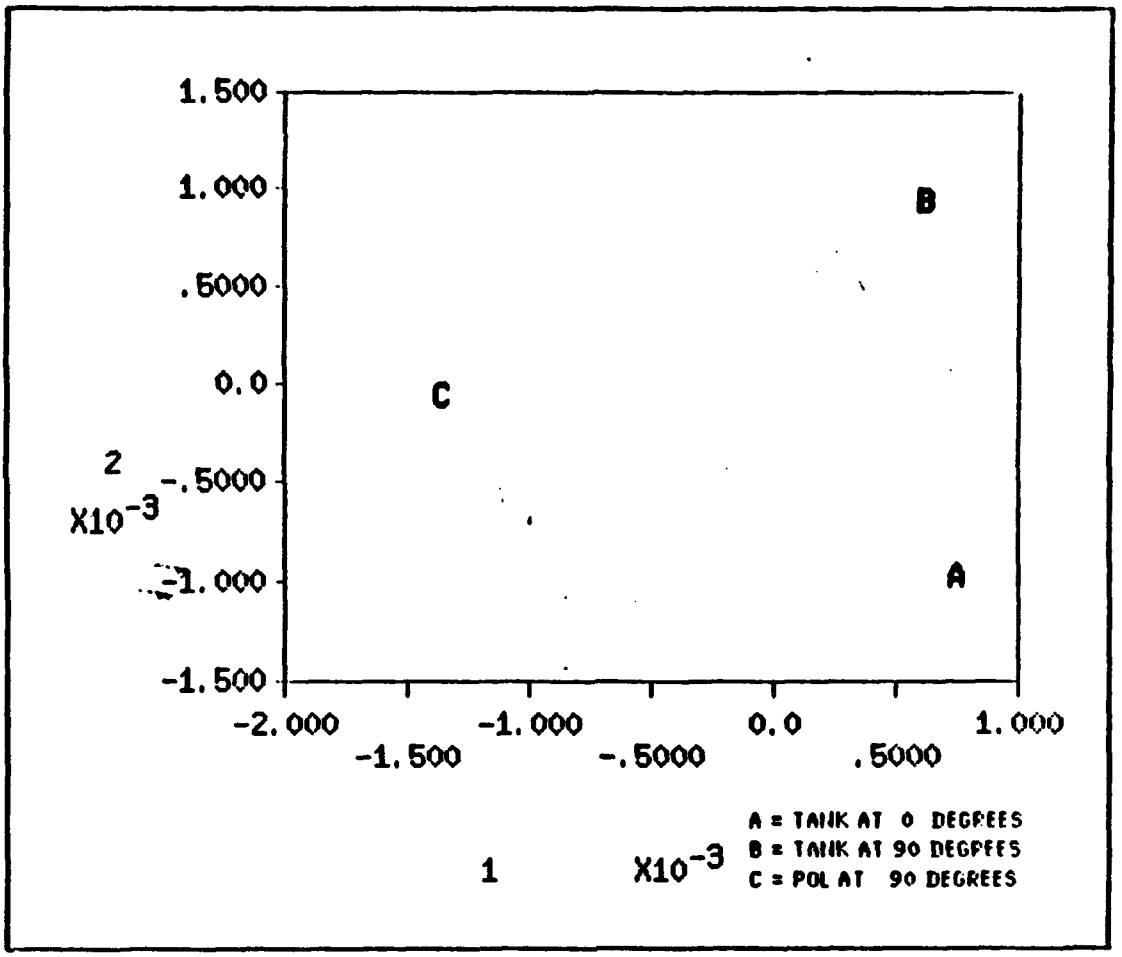

Figure 4.19 Scatter Plot of Feature Vectors for Rotation Changes

test verified that the feature vectors were PSRI. The next test was to determine the ability to separate classes of objects. Its ability to perform this separation has already been alluded to in the PSRI test.

4.3.1.2. Small Database Test Results

The first subset analyzed consisted of three classes which were to be recognized. With three classes, the dimension of the transformed feature space is guaranteed to be at most two; hence, the transformed feature vectors can be plotted in the 2D-plane to examine the separation of the classes. The original feature vector contained eleven moment invariants based on the silhouette of the object and eleven moment invariants based on

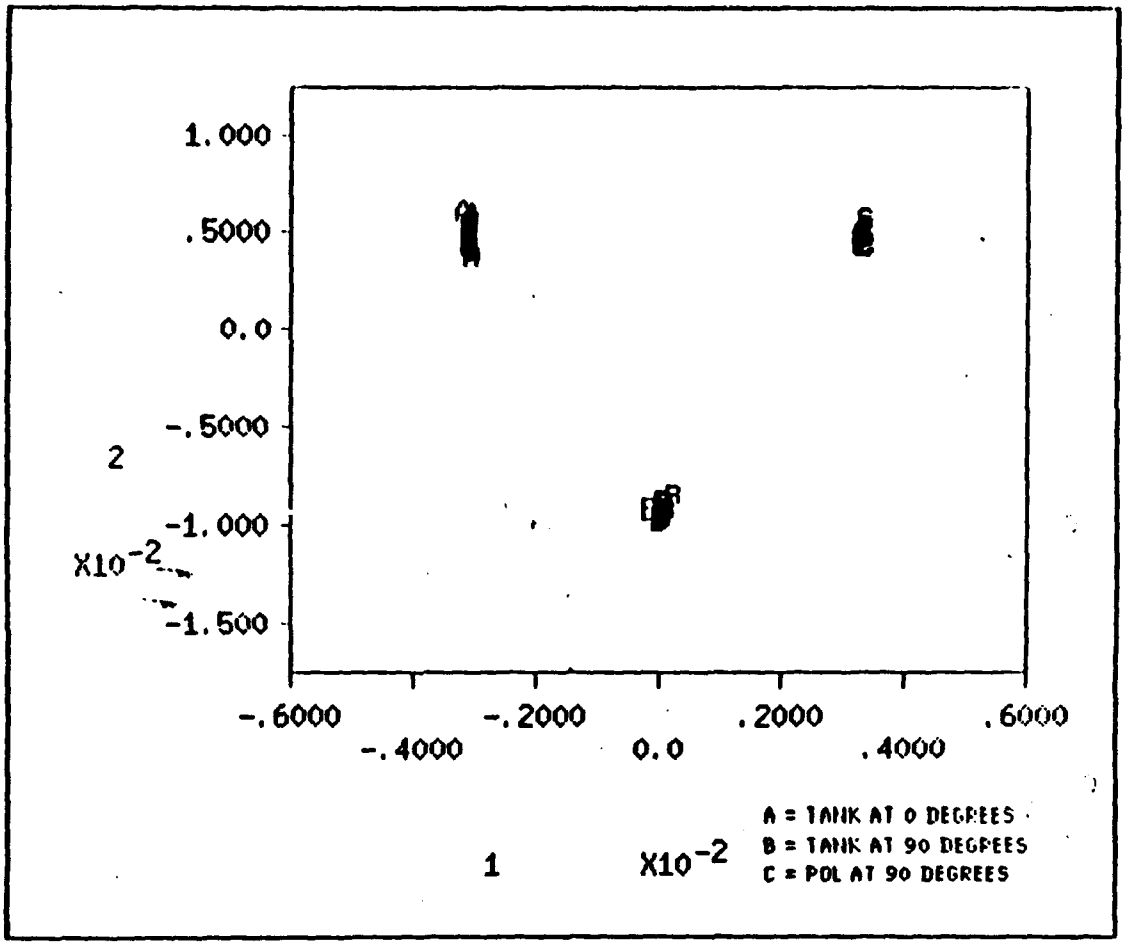

Figure 4.20 Scatter Plot of Feature Vectors for Rotation, Scale and Translation Changes

the border of the object. Figure 4.21 shows the scatter plot obtained when only the moment invariants for the silhouette were used. The eigenvalues determined during the computation of the Fisher linear discriminants were 2.6697 and 0.89855. As can be seen, the classes are somewhat separated but not as much as one would like. Figure 4.22 shows the scatter plot when only the moment invariants for the border are used. Here the eigenvalues are 1.2962 and 0.33213. Again, the classes are separated but not as much as desired. When both the silhouette moments and the border moments are used, the scatter plot of Figure 4.23 results. The eigenvalues for this case are 507.28 and 100.66. As can be seen, the use of both sets of moments provides exceptional separation of the classes indicating that classification of an unknown feature vector should be quite reliable.

These results are, however, misleading because the number of samples involved is insufficient.

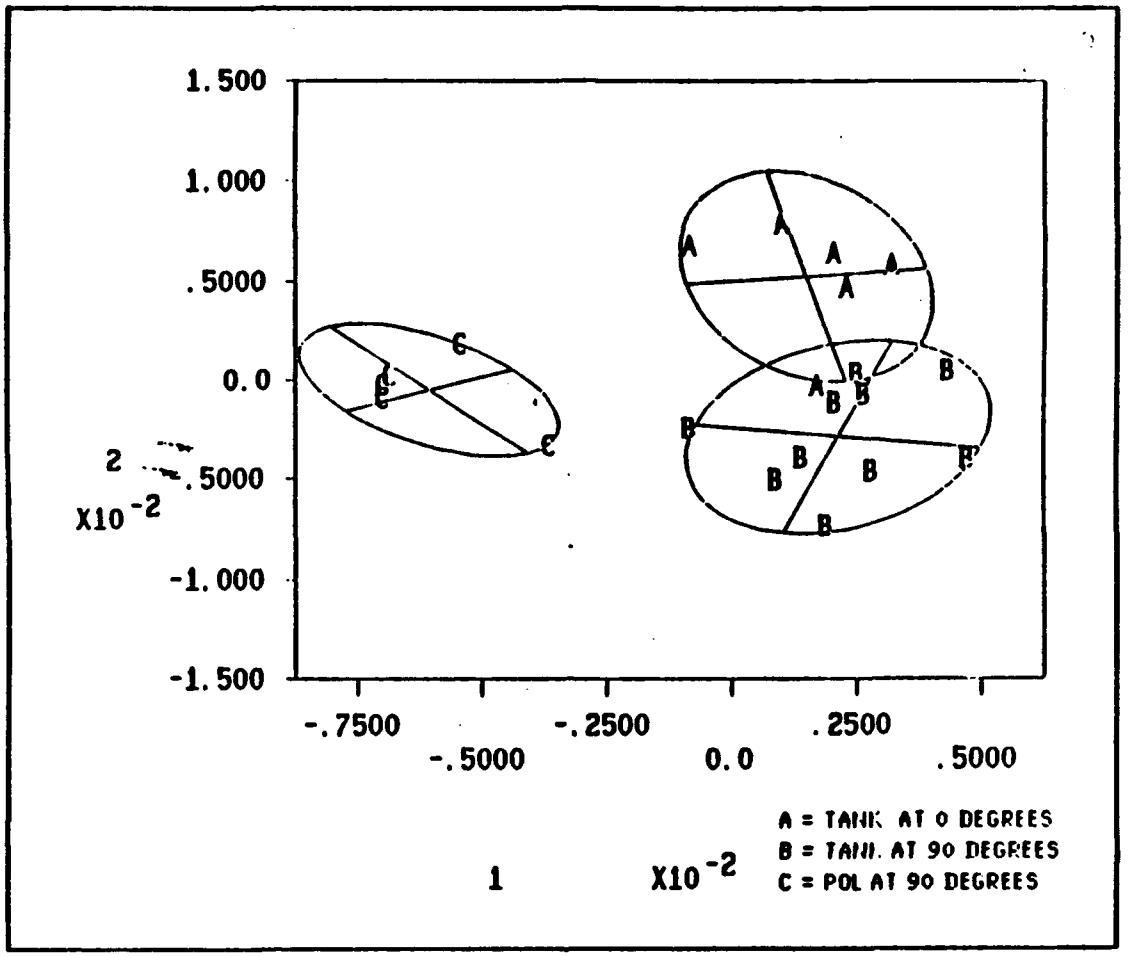

Figure 4.21 Scatter Plot of Small Database Using Silhouette Moments

The reason for the apparent excellent separation of the classes when both the silhouette and border moments are included is that the number of features being used is too large for the number of available samples. Foley [Foley72] discusses the relationship between the error rate for the design-set data and the ratio of samples per class to the number of features. The design-set error rate is the rate of errors when testing a classifier with the same data used to design it. Foley states that for a two-class problem when the ratio of total number of samples to the number of features is less than one, the design-set error rate is zero [Foley72:618]. In the above case, there are six front views of tanks, ten

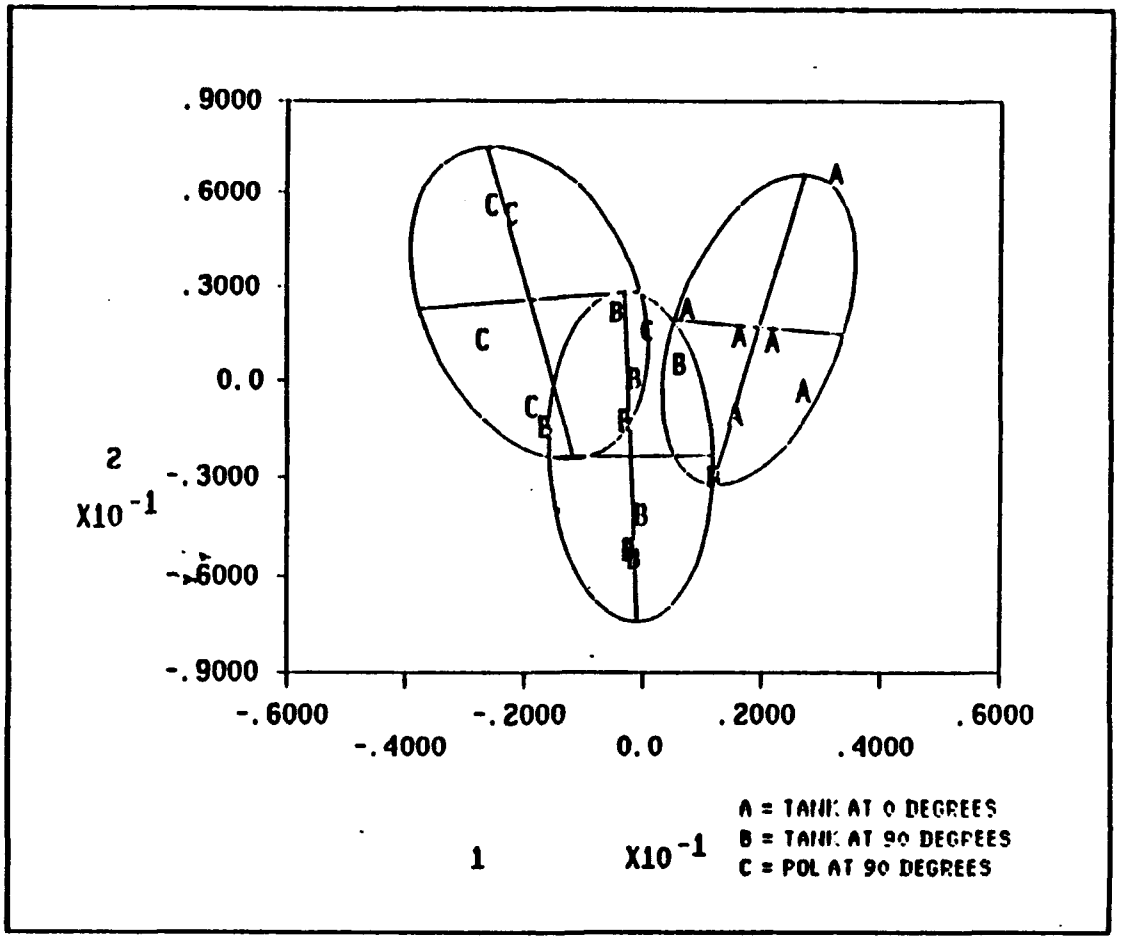

Figure 4.22 Scatter Plot of Small Database Using Border Moments

side views of tanks, and 5 side views of POLs; hence, when both silhouette and border moments are used, the ratio of total samples to features is $\frac{21}{22}$ which is less than one. When both sets of moments are used, the above rule is violated. As can be seen from Fig. 4.23, the design-set error rate is, indeed, zero. Foley shows that as the ratio of samples per class to the number of features approaches zero the design-set error rate approaches zero and that for ratios below three, the design-set error rate is extremely biased below the true error rate [Foley72:621]. Foley concludes that the number of samples per class to the number of features should be greater than three for the design-set error rate to be indicative of actual performance of the classifier [Foley72:623]. This is

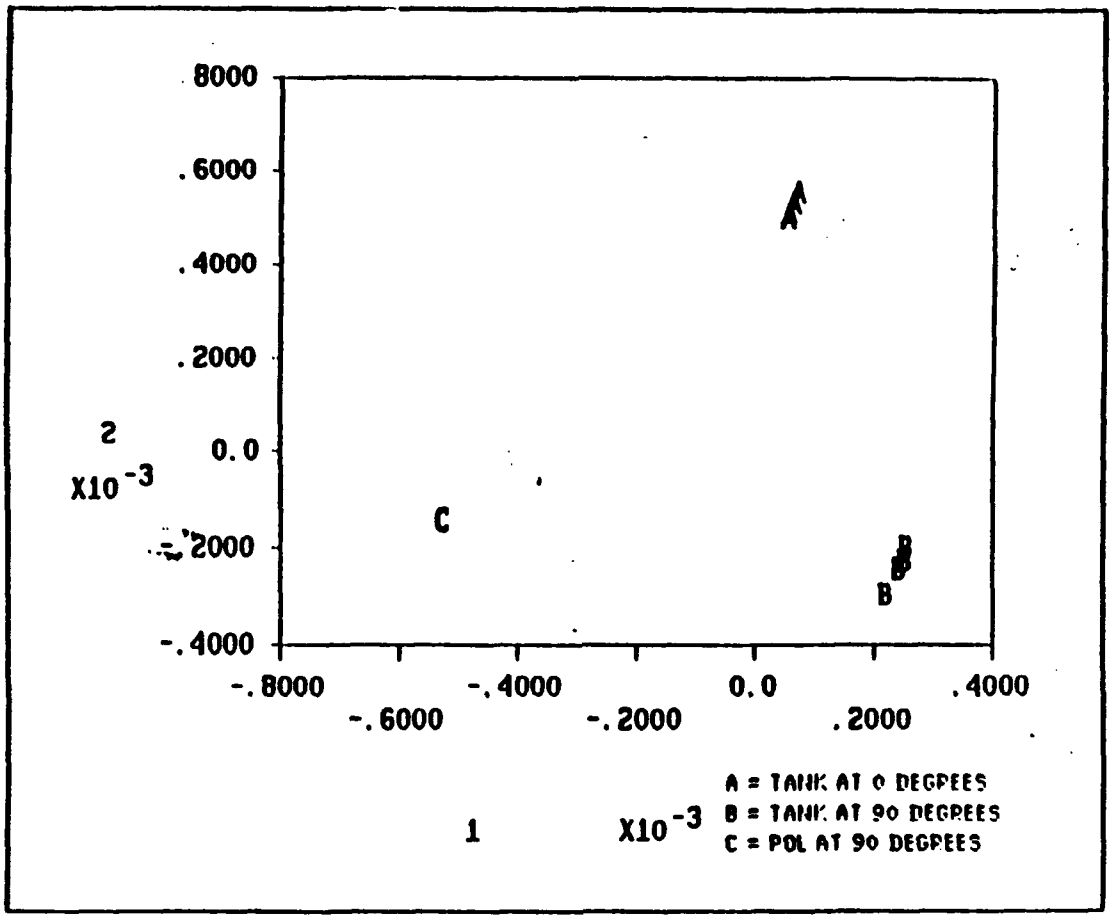

Figure 4.23 Scatter Plot of Small Database Using Silhouette and Border Moments

known as Foley's rule. In the above case, the ratio of samples per class to the number of features never exceeds one even when only a single set of moments is used; therefore, the design-set error rate will not indicate true classification ability. To avoid being directly affected by Foley's rule, all classifiers were designed using two-thirds of the sample vectors and tested with the remaining one-third. Hence, the design-set error rate was not used in evaluating the classifiers performance.

The nearest-neighbor classifier correctly classified four out of five test tanks and incorrectly classified both POLs used for testing. For the two POLs, the distance from the test vector and the nearest class was more than ten times as great as the average distance between a correctly classified test tank and the nearest class. Hence, the large

distance for the POLs indicates a poor match between the test vector and any of the reference classes. Apparently, more data is needed to properly characterize a POL in this feature space. Foley's rule would require 66 samples per class for reliable results.

4.3.1.3. Large Database Classification Results

The classification results for the larger database subset show promise; however, Foley's rule was not satisfied regarding the number of samples per class per feature. The larger database subset consisted of eleven classes as described in Table C.3 of Appendix C. Of these eleven classes, five classes were distinct (that is, not simply different aspects of the same object): tank, jeep, POL, 1.25 ton truck, and 2.5 ton truck. The eigenvalues from computing the Fisher linear discriminants are listed in Table 4.5. It is apparent from the eigenvalues that there are only about four dimensions of discrimination in the transformed feature vectors. Note an eigenvalue of zero indicates no variation between transformed vectors for the corresponding dimension. Since eleven classes were used to compute the Fisher linear discriminants, there could have been at most ten non-zero eigenvalues. Since it is difficult to display a scatter plot with four dimensions, the results of testing the classifier will be presented instead of scatter diagrams.

Table 4.5 Eigenvalues for Large Database

Rank	Eigenvalue	Rank	Eigenvalue
1	4.9755E+00	6	2.7946E-02
2	6.1017E-01	7	2.5266E-04
3	3.4125E-01	8	3.9190E-05
4	2.0287E-01	9	2.5925E-06
5	8.0128E-02	10	1.0586E-06

The classifier performed well for recognizing tanks and poorly for all other classes. Since there were only four samples for 1.25 ton trucks head-on and three samples for 1.25 ton truck side views, these classes were eliminated from the design-set. The confusion matrix of the test is shown in Table 4.6. Tanks were correctly recognized 76.5 percent of the time. The other classes were not reliably determined. It is likely that there was insufficient data to characterize the other classes. Certainly Foley's rule was not satisfied for any classes except tank where the total number of tanks used was 60. Foley's rule requires 66 in this case; hence, the tank class should be fairly well characterized.

Table 4.6 Nearest-Neighbor Confusion Matrix for Large Database. Reference Data (Columns) vs Test Data (Rows).

	Tank	Jeep	POL	Truck, 2.5 ton
Tank	13	3	1	0
Jeep	1	1	0	1
POL	3	0	0	2
Truck, 2.5 ton	0	0	1	1

The feature vectors were also classified using a neural network, in particular, a multilayer perceptron. The results of using this method for classification will be discussed next.

4.3.2. Multilayer Perceptron Classification

The multilayer perceptron was used to perform classification on the two database subsets described earlier. Each classifier was trained using two-thirds of the available samples and tested with the remaining one-third. The test results for the small database subset will be described first followed by a discussion of the results for the large database subset.

4.3.2.1. Small Database Results

The multilayer perceptron performed approximately the same for the small database subset as the classical nearest-neighbor. Two sets of data were classified using the multilayer perceptron. The first set of data consisted of the raw moment invariants, and the second set consisted of the Fisher linear discriminants for the training data. For the raw moment invariants, the system unambiguously classified all five test tanks as tanks. The two test POLs were split: one was unambiguously classified a POL and the other a tank. Similarly for the Fisher linear discriminants, four out of five tanks were correctly classified, and both POLs were incorrectly classified. With such a small test set, it is not possible to generalize on the differences between the results for the raw moments and the Fisher linear discriminants. Both seem to work equally well. The minimum within-class threshold was 0.80, and the maximum out-of-class threshold was 0.20. Table 4.7 lists the pertinent details regarding the training of the network. After testing the multilayer perceptron with the small database subset, the next step was to apply it to a larger database with more than two classes.

4.3.2.2. Large Database Results

The multilayer perceptron did not perform as well with the larger database as it did with the small database. The large data base included tanks at four different aspect angles, jeeps, POLs, and 2.5 ton trucks. The front views of the jeeps, POLs, and trucks were used for classification. Also included in testing the algorithm were several objects

Table 4.7 Multilayer Perceptron Training Data for Small Database

	Raw Moments	Fisher Linear Discriminants
Number of Classes	2	2
Number of Features	22	2
Training Rate (η)	0.25	0.25
Momentum (α)	0.9	0.9
First Layer Nodes	100	100
Second Layer Nodes	30	30
Output Error Criterion	0.01	0.01
Output Error Averaging Interval	200	200
Total Iterations	3700	444
Total Training Errors	144	39

generated by the relative range segmentation which did not correspond to vehicles. These objects were used as clutter to determine the ability of the network to reject objects for which it had not been trained. Again the multilayer perceptron was tested using two sets of data: the raw moment invariants and the Fisher linear discriminants. The training parameters and results are listed in Table 4.8, and the test results are summarized in Tables 4.9 and 4.10. There are several possible reasons for the poor performance of the network. Possible reasons include 1) insufficient data to characterize all classes of objects and 2) overlap of classes in the feature space. Each of these will be discussed separately.

It is likely that there was insufficient data to characterize the distributions of the classes being identified. The AFWAL database, as previously discussed, is heavily

Table 4.8 Multilayer Perceptron Training Data for Large Database

	Raw Moments	Fisher Linear Discriminants
Number of Classes	4	4
Number of Features	22	4
Training Rate (η)	0.25	0.25
Momentum (α)	0.9	0.9
First Layer Nodes	200	100
Second Layer Nodes	60	30
Output Error Criterion	0.05	0.28
Output Error Averaging Interval	200	200
Total Iterations	25650	3110
Total Training Errors	3553	848

skewed towards tanks. There are 274 tanks in the database out of 426 total targets. The large database subset also had a large number of tanks and much fewer objects for the other classes. The small number of objects for the other classes probably did not provide enough information about the distributions for the classes in order for the network to properly segment the decision space. As noted earlier, Foley's rule was only approximately satisfied for the class tank; all other classes had far fewer samples than required.

It is also likely that there was a significant amount of overlap between the classes in the feature space. Overlap in the feature space could result for several different reasons. First, some of the classes were very similar in shape. For example, the jeeps, POLs, and trucks were all basically rectangular objects. As previous segmentations have shown, the objects used for classification are very rough with very little detail (see Figures 4.8, 4.11,

Table 4.9 Multilayer Perceptron Confusion Matrix for Large Database and Clutter Data Using Raw Moment Invariants. Reference Data (columns) vs Test Data (Rows).

	Unambiguous Output				Ambiguous Output	
	Tank	Jeep	POL	Truck, 2.5 ton	Correct	Incorrect
Tank	19	1	0	0	1	1
Jeep	2	0	0	0	0	1
POL	0	0	0	0	0	2
Truck, 2.5 ton	0	0	0	1	0	1
Clutter	10	2	0	0	N-A	9

4.15, and 4.17). Thus, the jeeps, POLs, and trucks had generally similar shapes. Since only a few low order moments were used for shape description, fine shape detail is lost; hence, the jeeps, POLs, and trucks are bound to show overlap in this feature space.

A second reason for overlap in the feature space is due to noise in the shape segmentations. Noise in the shapes can come from changing aspect angles, occulting by other objects, carrier dropout, and so on. The noise results in the shapes of objects for a given class having a great deal of variation. This variation in shape can be translated into variation in the feature vectors resulting in the classes overlapping due to the spread of the distributions. It is likely that both of these factors played a part in the overlap of the classes.

Table 4.10 Multilayer Perceptron Confusion Matrix for Large Database and Clutter Data Using Fisher Linear Discriminants. Reference Data (columns) vs Test Data (Rows).

	Unambiguous Output				Ambiguous Output	
	Tank	Jeep	POL	Truck, 2.5 ton	Correct	Incorrect
Tank	13	3	0	0	1	5
Jeep	1	0	0	0	0	2
POL	0	0	0	0	0	2
Truck, 2.5 ton	1	0	0	0	0	1
Clutter	0	17	0	0	N-A	4

A comparison of the test results for the two types of input data, raw moment invariants and Fisher linear discriminants, indicates that the use of raw moment invariants is better. For the two categories with the most number of test samples (tanks and clutter) the performance was better using the raw moment invariants. Further research is required to determine if this is a general result and why.

4.4. Results Summary

In summary, two of the doppler segmentation algorithms developed were effective at detecting targets in an image. Both of the classification methods were useful for classifying targets; however, the multilayer perceptron outperformed the statistical nearest-neighbor classifier in every test. The classifiers were good at detecting tanks, but there

were insufficient samples to properly characterize the other classes. The next chapter will present conclusions and make recommendations for further research.

5. Conclusions & Recommendations

In this thesis, the problem of machine vision for the purpose of detecting and identifying targets in a tactical scenario was investigated. Doppler and relative range images were used for source imagery, and the targets of interest were M60 tanks, POLs, jeeps, 1.25 ton trucks, and 2.5 ton trucks.

5.1. Recommendations

The algorithms developed during the research show promise for locating and identifying tactical targets in a scene. There are six recommended avenues of further research to improve the confidence in these results and to improve the overall solution: identifying tactical targets.

1. Although two of the doppler segmentation algorithms were effective, an alternative method using the joint doppler-carrier histogram is recommended. As noted earlier, both methods will be only partially successful when an image contains targets moving both towards and away from the detector. Also, the Background Lobe Elimination method relies on two arbitrary thresholds for the carrier intensity and the main lobe width. A method which does not rely on arbitrary thresholds is preferred.

   The joint histogram provides all the information required to reliably segment a doppler image. From the joint histogram, it is immediately obvious which doppler returns are spurious and which are genuine. Note from Figure 4.H3204 that the spurious doppler returns are clumped together, the background lobe is a thick line, and any targets are represented by lines parallel to the background line. A doppler segmentation which finds the target lines should be developed based on these simple properties of the joint doppler-carrier histogram.

2. The approach used shows promise for classifying tactical targets in doppler and relative range imagery. In order to validate the approach, more data should be

obtained to allow the full characterization of all target classes of interest. With 22 features currently being used, Foley's rule requires 66 samples per class. There is currently only sufficient data to characterize M60 tanks. The recently acquired Army Night Vision Laboratory laser radar data may provide the needed additional data. Also, Rome Air Development Center (RADC) has laser radar data which may be useful for validating the algorithms developed in this thesis.

3. The moment invariants computed used only up to fourth order moments. If the current feature vector proves to be insufficient when more data is analyzed, then higher order moments should be included in the feature vector. The higher order moments will allow more precise shape definition. However, the addition of more features will require more samples per class to satisfy Foley's rule.

4. The shapes derived using the segmentation algorithms were very crude. Often a human observer could not determine the classification using the shape alone; however, when the observer examined all the images available for the scene, he could almost always properly identify the object. Feature vectors spanning the entire set of images available should be developed which do not rely entirely on the shape of the object. Three non-shape specific features are suggested: 1) estimation of the size of the object, 2) infrared intensity pattern over the object, 3) range variation over the object. Estimation of the size of the object will soon be possible when the new AM/FM laser radars are used because absolute range data will then be available. The other suggested features can be computed using the existing data.

5. Since each image type does not always give information about the class of an object (for example, a stationary target will have no doppler return), a heuristic algorithm should be developed which will combine all the available evidence from the various sensors to perform classification. This algorithm will be required to combine the evidence from multiple sensors. This avenue is now being pursued at the Air Force Institute of Technology by a doctoral student, Captain M. Roggemann

[Roggemann87]. It is recommended that evidential or probabilistic approaches such as those described in [Lee87] be investigated.

6. The multilayer perceptron and the back-propagation method should be investigated for the purpose of answering such questions as a) what is the best method of normalizing the input data and setting the connection weights to provide the fastest convergence, b) how does the number of nodes in the first and second layers effect the speed of convergence, c) how should the number of nodes in the first and second layers be selected, d) can the speed of convergence be improved by modifying the back-propagation method to not update weights when the actual output is greater than the desired output for an output node which is supposed to be active (this modification was noted in chapter 3), and e) will best classification performance be realized using the raw moment invariants or the Fisher linear discriminants?

5.2. Conclusions

The algorithms employed demonstrated classification of tactical targets from doppler and relative range imagery. The lack of sufficient data (according to Foley's rule) to characterize all the classes of interest, however, limits the conclusiveness of the results.

The segmentation algorithms were effective in finding the targets in doppler and relative range images. The new Optimum Thresholding and Background Lobe Elimination segmentation algorithms for doppler images proved to be highly reliable in locating the targets in an image finding a majority of the target pixels 87.2 and 88.9 percent of the time, respectively. The Background Lobe Elimination method was immune to spurious doppler returns due to carrier dropout. The relative range segmentation algorithm was effective in locating portions of the targets in an image. However, the resulting segmented shapes (doppler and relative range) often did not have sufficient shape to permit classification even by a human observer.

The Zernike moment invariants were shown to be useful in classifying the shapes of the detected objects. Both methods of classifying the feature vectors show promise. However, the lack of sufficient data to characterize the distributions for the classes prevents the assertion that these moment invariants are sufficient to discriminate the classes of interest. Tanks were correctly classified using the nearest-neighbor decision rule 76.4 percent of the time while the multilayer perceptron achieved 86.4 percent correct classification for tanks.

There are four primary contributions generated by this thesis. First, this research has produced two doppler radar segmentation algorithms which have proved to be highly reliable over a wide range of target and background conditions. The Background Lobe Elimination method proved to be effective even in the presence of noise due to severe carrier dropout. The second contribution is the application of neural networks to real world sensor data. It has been shown that the multilayer perceptron can be effectively used for automatic target recognition. Third, this research has shown that the multilayer perceptron performs classification better than the best statistical classifier possible. The neural network has significant advantages over the traditional statistical classifiers. First, the neural network is inherently extensible through the addition of more nodes, and the addition of more nodes will not significantly degrade performance because the nodes in each layer operate in parallel. Another advantage of the neural network is fault tolerance. Since the decision rule implemented by the neural network is distributed through the network in the connection weights and thresholds, the system should degrade gracefully as nodes and weights are lost. Also, neural networks can implement highly complex partitions of the feature space automatically through the training process. Finally, neural networks are computationally quick because calculations in each layer occur in parallel. The fourth contribution of this research is the demonstration of a classification algorithm which performed as well as a human observer could using the same available information.

With the results of this thesis, another researcher has available segmentation algorithms for both doppler and relative range images in a software environment which is easy to use and provides powerful and diverse tools. The next step in the research is to combine these segmentations with information from the other sensors to improve the classification performance. In particular, the different modality sensors, the passive visible and infrared, should be utilized.

The software environment developed provides a rich set of tools for the acquisition and manipulation of images and regions extracted from images. The next researcher can start working immediately at a higher level of image processing rather than developing a basic image processing facility. The way is now clear for a truly multisensor approach using the AFWAL database and the software and algorithms developed at AFIT.

Appendix A : Computing Fisher Linear Discriminants

Introduction

The use of Fisher Linear Discriminants (also known as principal components analysis) is a standard technique in pattern analysis which is applied to a feature space in order to make classification easier. The primary advantage of principal component analysis is a reduction in the dimensionality of the decision space with a minimal reduction in classification performance. That is, the features vectors are transformed from d dimensions to $c-1$ dimensions where c is the number of classes to recognize. Although this transformation maps many vectors in the original space to a single vector in the decision space, the performance in the new space is close tc optimal and is optimal when the original feature vectors are Gaussian random variables with equal covariances [Duda73:118].

The use of ILS for principal components analysis consists of five primary steps: 1) read the feature vectors from text files into ILS format files, 2) normalize the data, 3) compute the Fisher Linear Discriminants, 4) transform the feature vectors, and 5) dimensionality reduction of the transformed vectors. Each step will be described separately, but first some ILS terminology will be defined.

ILS Terminology

ILS terms to be discussed are primary and secondary files, feature data, records, elements, items, and fields. ILS uses files for input and output for almost all operations. The primary files are normally the input data for an operation while the secondary files are used to output data. ILS file names consist of two parts. The first part is up to four alphabetic characters. The second part is a number between 1 and 9998. For a given ILS session, the alphabetic prefix is constant while the numbers are changed to send data to different files. The file names are defined using the FIL command. When a command

writes data to a secondary file, the file must have been previously opened. To open a file, use the OPN command.

ILS can work with several types of data. The type of data used for feature vector analysis is called feature record data which is a vector of real numbers. In ILS terminology, each vector is called an item, and the individual components of each vector are called elements. An ILS record can contain more than one item (vector); however, normally the number of records is equal to the number of items (vectors) in a file.

Each feature record in ILS has several fields associated with it that describe the data contained in the record. For principal components analysis, only two fields are used: field-1 and field-2. The user decides what the two fields mean. Each field can contain up to eight alphanumeric characters. Normally, one of the two fields will indicate which class the feature vector comes from. This use allows the BPA command to perform classification of test vectors given a set of reference vectors.

In this section, essential ILS terminology has been defined. In the next section, the method for reading textual data into ILS will be described.

Reading Text Files from ILS

The first step is to read the feature vectors from a set of text files into ILS feature record files. The text files must be in the following format. For each vector, there are three parts 1) field-1, 2) field-2, and 3) the components of the vector. The fields must each start on the beginning of a new line in the text file. ILS will read only the first eight characters from each line. The component data must begin on or after the line following field-2. The component data can have any columnar spacing and valid FORTRAN floating point format. When using Ada, the numbers can simply be printed out in the usual exponential notation.

To read the text files the WRT command is used. The WRT command reads text data from the file WRTIN.DAT; hence, each text file must be copied to that name as it is

read in. This requirement makes a multi-window workstation very useful when running ILS since one window can run ILS while the other window is used to copy the input data to the file WRTIN.DAT. The WRT command is an exception to the above rule about primary and secondary files. Since the input data comes from a file outside ILS, the output is sent to the primary file instead of the secondary file. The command to read the text file is WRT FF1,(dimension) where (dimension) is a number indicating the dimension of the feature vectors. It is recommended that each text file contain feature vectors for only one class (in the case of reference data) and that the ILS files be numbered consecutively for each group of reference data. The reason for this is that other ILS commands use multiple files which must be numbered sequentially.

The feature vectors have now been read into files which can be easily manipulated using ILS commands. The next step in the process is to normalize the feature vectors.

Feature Vector Normalization

Feature vector normalization consists of insuring that the mean of each component is zero and the variance is one across all reference vectors (not within a given class). This process insures that scaling of the axes is uniform for all components of the feature vectors; hence, no single component can dominate the calculation simply due to a difference in scaling. The normalization process consists of four steps: 1) combining the vectors into one file, 2) computing the mean and standard deviation vectors, 3) translating the feature vectors, and 4) scaling the feature vectors. The feature vectors are combined into a single file using the TRE command. The mean and standard deviation are computed using SME 3,1.

To translate and scale the data the BOP command is used. The primary file is the original feature vector file for a given class; the primary B file is the file produced by the SME 3,1 command; and the secondary file is where the translated feature vectors for a given class will be sent. As implied, each class of feature vectors is separately translated

and scaled. The command to translate the data is BOP S,,,2. When scaling the data, the primary file is the translated file for a given class; the primary B file is the product of SME 3,1; and the secondary file receives the scaled data. The command to scale the data is BOP D,,2,2.

The feature vectors have now been normalized, so that the scaling on each axis is the same; hence, no arbitrary feature can dominate the feature vector simply due to a scale difference between the features. The next section will describe how the Fisher Linear Discriminants are computed from the normalized feature vectors.

Computing Fisher Linear Discriminants

The Fisher linear discriminants are actually a transformation matrix which rotates and scales the original feature space into another vector space with a reduced dimensionality while maintaining classification capability. The SME command is used to compute the Fisher linear discriminants. The primary file is set to the file number for the first class (every class file must be sequentially numbered following the first class). The secondary file receives the Fisher linear discriminant data. The command to perform the analysis is SME 6,(num classes) where (num classes) is a number representing the number of classes being analyzed. This command produces the within-class scatter matrix, between-class scatter matrix, eigenvalues, and eigenvectors for the transformation. Theoretically, the maximum number of non-zero eigenvalues is one less than the number of classes; however, due to the finite amount of data and limited precision of the computer, the other eigenvalues will be very small but not zero. The number of non-zero eigenvalues indicates the dimensionality of the transformed vector space. For example, if only the first two eigenvalues are large compared to the rest, then the decision space is a plane. See Duda and Hart *Pattern Classification and Scene Analysis* for a complete explanation of the Fisher linear discriminants.

In this section, the method for computing the Fisher linear discriminants has been

described. The next section will discuss how the transformation is applied to the feature vectors.

Feature Vector Transformation

The Fisher linear discriminants are computed using the PCO command. The primary file is the first class of the normalized feature vectors. The primary B file is the file produced by the SME 6 command. The transformed data for each class is placed in a separate file. The secondary file receives the first class and each succeeding file receives the following classes. These secondary files must be opened prior to executing the PCO command. The files can be opened using the following concatenated command string FIL Snnnn/OPN S(num classes) where nnnn is the file number of the first file to receive the first class and (num classes) is the same as before. Once the primary and secondary files are set use the following command to transform the feature vectors PCO „(num classes) where (num classes) is the same as before. If the transformed data is two dimensional (according to the eigenvalues), it can be effectively plotted now using the command PLR S. The PLR command prompts for the required information. In order to save the plotted graphs on disk use ASG GO1 before executing the PLR command. Then the plots will go to the file GRAPH.DAT as well as being displayed on the screen.

Feature Vector Dimensionality Reduction

If it is desired to later use the transformed feature vectors for classification tests, the dimension of the transformed vectors must be reduced to the dimension indicated by the eigenvalues. The dimension must be reduced because the higher dimensions contain spurious values which prevent the classification utility BPA from working correctly. The dimension of the feature vectors can be reduced using the MRE command. This command is interactive. When requested for the elements (components) to be moved use 1, -(dimension) where (dimension) is a number indicating the dimension of the transformed

vector space. The minus sign indicates that all elements from 1 to (dimension) inclusive are to be moved.

Conclusion

The Fisher linear discriminants provide a powerful technique for reducing the dimensionality of the feature space while preserving for the most part the classification effectiveness of the original feature space. ILS can work with feature spaces up to 32 dimensions under the current release (for 32 bit systems). See the descriptions of the commands SME and PCO in ILS Volume II and the application note on Pattern Analysis in ILS Volume III for more information.

Appendix B : Using ILS for Classifying Feature Vectors

Introduction

ILS can be used to classify feature vectors using a variety of classification decision rules. The use of ILS for implementing a simple nearest neighbor decision rule will be described here. The decision rule to be used consists of computing the distances between a test vector and the mean for each class. The class mean which is closest to the test vector is selected as the class of the test vector. Other more complex decision rules are available under ILS using the command BPA. It is assumed that the feature vectors have been preprocessed using PCO and that the reader is familiar with ILS terminology (For an explanation of these, see Appendix A).

Using ILS for nearest neighbor classification requires three steps: 1) computation of reference means, 2) transformation of raw feature vectors into the decision space, and 3) classification using BPA.

Reference Means Computation

The reference means are computed using the SME command. The primary file is set to the first class. The secondary file receives the computed means. The command used is SME 1,(num classes) where (num classes) is the number of classes being analyzed.

Test Vector Transformation

Once the reference means are computed, the test vectors must be transformed in the same way the reference vectors were transformed using PCO. After the test vectors have been transformed, the classification can be performed.

Test Vector Classification

The ILS command BPA performs the classification. The output of BPA is normally sent to the lineprinter (which is actually a file SYS$PRINT.DAT on the VAXstation system. This file is not accessible while ILS is running; hence, it is normally desirable to redirect the output from this file to the screen). To send the output of BPA to the screen instead of the lineprinter, use the command ASG LP6. To restart sending output to the lineprinter use the command ASG LP7.

The command BPA will also output distances and the confusion matrix to ILS files if desired. If the distances are to be saved, the secondary file should be opened using OPN S. If the confusion matrix is to be save, the secondary B file should be opened using OPN SB. The primary file is set to the file containing the means of the reference vectors, i.e., the product of the SME 1,(num classes) command. The primary B file is set to the file number of the first set of test vectors (normally from a given class).

The class is determined by either field-1 or field-2. The user is prompted for which field to use and which characters within the field to use. Once the primary file has been set and the secondary files opened, the command to start classification is BPA. Select option 1 for Euclidean distance measure and answer all questions from BPA.

Conclusion

The ILS command BPA provides an effective method for testing the classification effectiveness of a given feature space. Other more powerful techniques are also available using the BPA command. See the description of BPA in ILS Volume II and the application note on Pattern Analysis in ILS Volume III for more information.

Appendix C : Source Data Used for Segmentation and Classification

This appendix consists of a set of tables which describe the details of the source data used for testing the segmentation and classification algorithms.

Table C.1 Target Classes for Small Database

Class	Aspect Angle (deg)	Training Samples	Testing Samples
Tank	0	4	2
	90	7	3
POL	90	3	2

The following table lists the images used for the small database. The names listed are the first five letters of the file names used to store the images in the AFWAL database.

Table C.2 Small Database Source Images

Doppler	Relative Range
D3028	R3033
D3033	R3083
D3035	R3090
D3042	R3190
D3051	R3195
D3066	
D3074	
D3088	
D3090	
D3195	
D3197	

The following table gives the target classes used for the large database subset. For classes where the number of training samples is zero, the test vectors were used to determine the ability of the classification algorithm to reject untrained classes.

Table C.3 Target Classes for Large Database

Class	Aspect Angle (deg)	Training Samples	Testing Samples
Tank	0	14	6
	45	8	4
	90	8	4
	315	13	3
Jeep	All	6	3
POL	0	4	2
	90	0	3
Truck, 1.25 ton	0	0	4
	45	0	3
Truck, 2.5 ton	0	4	2
	90	0	2

Table C.4 Source Images for Large Database

3100 Series		3200 Series	3300 Series
D3104	D3112	D3200	D3302
D3116	D3117	D3203	D3308
D3126	D3128	D3204	D3311
D3135	D3136	D3208	D3313
D3140	D3142	D3209	D3314
D3145	D3147	D3211	D3322
D3149	D3153	D3215	D3324
D3157	D3161	D3238	D3326
D3164	D3165	D3257	D3327
D3166	D3178	D3258	D3328
D3179	D3180	D3268	D3330
D3192		D3278	D3332
		D3286	D3333
		D3288	D3334
			D3338
			D3339
			D3344
			D3345

The following four tables list the AFWAL database doppler images used to evaluate the performance of the segmentation algorithms. A total of 180 doppler images were evaluated.

Table C.5 Series 3000 Doppler Images Segmented

D3026	D3028	D3030	D3033
D3035	D3038	D3040	D3042
D3045	D3047	D3051	D3053
D3055	D3057	D3061	D3064
D3066	D3069	D3071	D3074
D3077	D3078	D3079	D3080
D3081	D3085	D3087	D3088
D3090	D3092	D3096	D3098

Table C.6 Series 3100 Doppler Images Segmented

D3100	D3104	D3105	D3112
D3113	D3114	D3116	D3117
D3136	D3137	D3140	D3141
D3142	D3149	D3153	D3157
D3158	D3159	D3161	D3162
D3165	D3166	D3170	D3171
D3174	D3175	D3176	D3178
D3179	D3180	D3181	D3182
D3183	D3184	D3185	D3192
D3193	D3195	D3196	D3197
D3199			

Table C.7 Series 3200 Doppler Images Segmented

D3200	D3201	D3202	D3203
D3204	D3205	D3206	D3207
D3208	D3209	D3211	D3213
D3214	D3215	D3217	D3218
D3219	D3225	D3226	D3227
D3230	D3231	D3232	D3233
D3234	D3235	D3237	D3238
D3239	D3241	D3242	D3244
D3245	D3246	D3247	D3249
D3250	D3251	D3255	D3256
D3257	D3258	D3268	D3269
D3278	D3279	D3280	D3281
D3282	D3285	D3286	D3287
D3288	D3289	D3291	D3292
D3293	D3294	D3295	D3297
D3298	D3299		

Table C.8 Series 3300 Doppler Images Segmented

D3300	D3301	D3302	D3303
D3304	D3305	D3306	D3307
D3308	D3309	D3310	D3311
D3313	D3314	D3315	D3319
D3320	D3321	D3322	D3324
D3326	D3327	D3328	D3329
D3330	D3331	D3332	D3333
D3334	D3335	D3338	D3339
D3340	D3343	D3344	D3345
D3346	D3347	D3348	D3349
D3350	D3351	D3352	D3353
D3354			

Table C.9 Relative Range Images Used for Evaluating Segmentation Algorithm

R3023	R3024	R3025	R3026
R3027	R3028	R3029	R3030
R3031	R3032	R3033	R3034
R3035	R3036	R3037	R3038
R3039	R3040	R3041	R3042
R3043	R3044	R3045	R3046
R3047	R3048	R3049	R3050
R3051	R3052	R3053	R3054
R3055	R3056	R3057	R3058
R3059	R3060	R3061	R3062
R3063	R3064	R3065	R3066
R3067	R3068	R3069	R3070
R3071	R3072	R3073	R3074
R3075	R3076	R3077	R3078
R3079	R3080	R3081	R3082
R3083	R3084	R3085	R3086
R3087	R3088	R3089	R3090
R3091	R3092	R3093	R3094
R3095	R3096	R3097	R3098
R3099			

Appendix E : Zernike Moment Invariants

The following eleven moment invariants were used as shape descriptors for the regions detected. The moment invariants were derived by Teague from Zernike moments in [Teague80]. The moments μ'_{pq} are derived for an object using equations (2.3) and (2.4). These moments are position, scale and rotation invariant.

$$S_1 = 3[2(\mu'_{20}+\mu'_{02})-1]/\pi \tag{E.1}$$

$$S_2 = 9[9\mu'_{20}-\mu'_{02})^2 + 4(\mu'_{11})^2]/\pi^2 \tag{E.2}$$

$$S_3 = 16[(\mu'_{03}-3\mu'_{21})^2 + (\mu'_{30}-3\mu'_{12})^2]/\pi^2 \tag{E.3}$$

$$S_4 = 144[(\mu'_{03}+\mu'_{21})^2 + (\mu'_{30}+3\mu'_{12})^2]/\pi^2 \tag{E.4}$$

$$S_5 = \frac{13824}{\pi^4}\{(\mu'_{03}-3\mu'_{21})(\mu'_{03}+\mu'_{21})[(\mu'_{03}+\mu'_{21})^2-3(\mu'_{30}+\mu'_{12})^2] - (\mu'_{30}-3\mu'_{12})(\mu'_{30}+\mu'_{12})[(\mu'_{30}+\mu'_{12})^2-3(\mu'_{03}+\mu'_{21})^2]\} \tag{E.5}$$

$$S_6 = \frac{864}{\pi^3}\{(\mu'_{02}-\mu'_{20})[(\mu'_{03}+\mu'_{21})^2-(\mu'_{30}+\mu'_{12})^2]+4\mu'_{11}(\mu'_{03}+\mu'_{21})(\mu'_{30}+\mu'_{12})\} \tag{E.6}$$

$$S_7 = 25[(\mu'_{40}-6\mu'_{22}+\mu'_{04})^2 + 16(\mu'_{31}-\mu'_{13})^2]/\pi^2 \tag{E.7}$$

$$S_8 = 25\{[4(\mu'_{04}-\mu'_{40}) + 3(\mu'_{20}-\mu'_{02})]^2 + 4[4(\mu'_{31}+\mu'_{13})-3\mu'_{11}]^2\}/\pi^2 \tag{E.8}$$

$$S_9 = 5[6(\mu'_{40}+2\mu'_{22}+\mu'_{04}) - 6(\mu'_{20}+\mu'_{02}) + 1]/\pi \tag{E.9}$$

$$S_{10} = \frac{250}{\pi^3}((\mu'_{40}-6\mu'_{22}+\mu'_{04}) \times 14[(\mu'_{04}-\mu'_{40}) + 3(\mu'_{20}-\mu'_{02})]^2 - 4[4(\mu'_{31}+\mu'_{13})-3\mu'_{11}]^2\} - 16[4(\mu'_{04}-\mu'_{40}) + 3(\mu'_{20}-\mu'_{02})] \times[4(\mu'_{31}+\mu'_{13}) - 3\mu'_{11}](\mu'_{31}-\mu'_{13})) \tag{E.10}$$

$$S_{11} = \frac{30}{\pi^2}\{[4(\mu'_{04}-\mu'_{40}) + 3(\mu'_{20}-\mu'_{02})](\mu'_{02}-\mu'_{20})$$

$$+ 4\mu'_{11}[4(\mu'_{31}+\mu'_{13}) - 3\mu'_{11}]\} \qquad \text{(E.11)}$$

Bibliography

Abu-Mostafa, Yaser and Demetri Psaltis. "Recognitive Aspects of Moment Invariants," *IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-6:*698-706 (November 1984).

Andrews, Harry C. *Introduction to Mathematical Techniques in Pattern Recognition.* New York: Wiley-Interscience, 1972.

Duda, Richard O. and Peter E. Hart. *Pattern Classification and Scene Analysis*. New York, London, Sydney, and Toronto: John Wiley & Sons, 1973.

Dudani, S.A. *et al.* "Aircraft Identification by Moment Invariants," *IEEE Transactions on Computers, C-26:*39- 45 (January 1977).

Foley, Donald H. "Considerations of Sample and Feature Size," *IEEE Transactions on Information Theory, IT-18:* 618-626 (September 1972).

Gonzalez, Rafael C. and Paul Wintz. *Digital Image Processing*. London, Amsterdam, Don Mills, Ontario, Sydney, and Tokyo: Addison-Wesley Publishing Company, 1977.

Lee, Tong *et al.* "Probabilistic and Evidential Approaches for Multisource Data Analysis," *IEEE Transactions on Geoscience and Remote Sensing, GE-25:*283-293 (May 1987). Lippmann, Richard P. "An Introduction to Computing with Neural Networks," *IEEE ASSP Magazine, 4:*4-22 (April 1987).

Melsa, James L. and David L. Cohn. *Decision and Estimation Theory*. New York: McGraw-Hill Book Compar.y, 1978.

Parker, David B. *Second Order Back Propagation: Implementing an Optimal O(n) Approximation to Newton's Method as an Artificial Neural Network.* To be published in *Computer* September 16, 1987. Menlo Park, CA.

Pavlidis, Theodosios. *Structural Pattern Recognition.* Berlin, Heidelberg, and New York: Springer-Verlag, 1977.

Roggemann, Capt. Michael. *General Principles for Multisensor Target Recognition.* PhD dissertation prospectus. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, September 1987.

Ross, Sheldon. *A First Course in Probability*. New York: Macmillan Publishing Co., Inc., 1976.

Rumelhart, D.E. *et al. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations.* Cambridge MA: MIT Press, 1986.

Target and Clutter Signatures from Multifunctional CO_2 Laser Radar. Company Brochure. LTV Aerospace and Defense, Dallas TX, undated.

Teague, Michael Reed. "Image Analysis via the General Theory of Moments," *Journal of the Optical Society of America, 70:*920-930 (August 1980).

Tong, Carl W. *Target Segmentation and Image Enhancement through Multisensor Data Fusion.* MS thesis, AFIT/GE/ENG/86D-55. School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB Ohio, December 1986.